International Journal of Information Movement ***Vol. 9 Issue I*** ***(May 2024)***
Website: www.ijim.in ***ISSN: 2456-0553 (online)*** **Pages 20-29**

DESIGNING A MODERN LIBRARY WEBSITE USING TAILWIND CSS

Lakhpat Singh
Librarian
Maharaja Agrasen Mahavidyalaya,
Jagadhri (Haryana)
Email: bablukuk@gmail.com

Abstract: This paper explores the process of designing a modern library website using Tailwind CSS, a utility-first CSS framework. Tailwind CSS enables rapid development by leveraging a comprehensive set of pre-designed utility classes, facilitating a highly customizable and responsive design approach. The project focuses on enhancing user experience through intuitive navigation, accessible information architecture, and visually appealing interface elements. By adopting Tailwind CSS, the design achieves a balance between aesthetic appeal and functionality, catering to diverse user needs in accessing library resources efficiently. This abstract highlights the strategic implementation of Tailwind CSS in creating a user-centric library website that prioritizes usability and accessibility.

Keywords: Website, Tailwind, Web-design, Open-Source Software, CSS, Library, Html

1.0 Introduction

Designing a modern library website involves integrating contemporary design principles with practical functionality. Tailwind CSS, a utility-first CSS framework, offers a streamlined approach to achieve this balance effectively. By utilizing its extensive library of utility classes, designers can rapidly prototype and refine UI components without writing custom CSS, ensuring consistency and scalability across the site. The benefits of Tailwind CSS in creating a visually cohesive and user-friendly library website. Emphasizing responsiveness and accessibility, the framework enables developers to focus on enhancing user experience through intuitive navigation, clear content presentation, and seamless integration of interactive elements. Website UI will be created a very beautiful and responsive with adding some few tailwind CSS classes.

2.0 What is Tailwind CSS?

Tailwind CSS is a utility-first CSS framework that streamlines the process of building and styling websites by providing a set of low-level utility classes. Unlike traditional CSS frameworks that use pre-designed components, Tailwind CSS encourages a more flexible and customizable approach. It includes a comprehensive collection of utility classes that directly apply styles like margins, paddings, colors, and more, directly in the HTML markup. This approach allows developers to quickly prototype designs, maintain consistency, and easily customize styles without writing additional CSS. Tailwind CSS is particularly favored for its efficiency in handling complex layouts and its ability to scale with large projects. Rapidly build modern websites without ever leaving the HTML.

A utility-first CSS framework packed with classes like flex, pt-4, text-center and rotate-90 that can be composed to build any design, directly in your markup.

3.0 Brief History of Tailwind CSS

Tailwind CSS was initially developed by Adam Wathan, and the first version was released back on the 1st of November, 2017. Tailwind CSS is a utility-first CSS framework for building custom user interfaces rapidly and efficiently. It is an inline styling used to achieve a sleek interface without writing code for your own CSS. Tailwind CSS offers customizability and flexibility to transform the appearance and feel of the elements. Tailwind CSS is not the first utility-first CSS library, but it is one of the most popular and light ones. It is a highly customizable, low-level CSS framework, and it provides all the building blocks the developer needs to build a fantastic interface for any website.

Tailwind CSS is an open-source CSS framework. The main feature of this library is that, unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables. Instead, it creates a list of "utility" CSS classes that can be used to style each element by mixing and matching.

For example, in other traditional systems, there would be a class message-warning that would apply a yellow background color and bold text. To achieve this result in Tailwind, one would have to apply a set of classes created by the library: bg-yellow-300 and font-bold.

4.0 Major Features of Tailwind CSS

4.1 Never ship unused CSS again: Tailwind automatically removes all unused CSS when building for production, which means your final CSS bundle is the smallest it could possibly be. In fact, most Tailwind projects ship less than 10kB of CSS to the client.
4.2 Responsive everything: Wrestling with a bunch of complex media queries in your CSS sucks, so Tailwind lets you build responsive designs right in your HTML instead. Throw a screen size in front of literally any utility class and watch it magically apply at a specific breakpoint.
4.3 Build whatever you want: Because Tailwind is so low-level, it never encourages you to design the same site twice. Even with the same color palette and sizing scale, it's easy to build the same component with a completely different look in the next project.
4.4 An API for design system: Utility classes help you work within the constraints of a system instead of littering your stylesheets with arbitrary values. They make it easy to be consistent with color choices, spacing, typography, shadows, and everything else that makes up a well-engineered design system.
4.5 Hover and focus states: Want to style something on hover? Stick hover: at the beginning of the class you want to add. Works for focus, active, disabled, focus-within, focus-visible, and even fancy states we invented ourselves like group-hover.
4.6 Don't worried about duplication: If you're repeating the same utilities over and over and over again, all you have to do is extract them into a component or template partial and boom — you've got a single source of truth so you can make changes in one place.
4.7 Extend it, tweak it, change it: Tailwind includes an expertly crafted set of defaults out-of-the-box, but literally everything can be customized from the color palette to the spacing scale to the box shadows to the mouse cursor. Use the tailwind.config.js file to craft your own design system, then let Tailwind transform it into your own custom CSS framework.
4.8 World-class IDE integration: Worried about remembering all of these class names? The Tailwind CSS IntelliSense extension for VS Code has you covered. Get intelligent autocomplete suggestions, linting, class definitions and more, all within your editor and with no configuration required.
4.9 Move even faster with Tailwind UI: Tailwind UI is a collection of beautiful, fully responsive UI components, designed and developed by us, the creators of Tailwind CSS. It's got hundreds of ready-to-use examples to choose from, and is guaranteed to help you find the perfect starting point for what you want to build.

5.0 Advantage of using Tailwind in Library's websites

Using Tailwind CSS in library websites offers numerous advantages that cater specifically to the needs of libraries and their diverse user bases. Here are some key advantages:

5.1 Customization: Tailwind CSS provides extensive utility classes that allow developers to customize every aspect of the website's design. This flexibility ensures that libraries can maintain their unique branding and cater to specific user preferences without being constrained by pre-defined components.
5.2 Responsive Design: Tailwind CSS simplifies the creation of responsive layouts by providing utility classes that directly target different screen sizes. This ensures that library websites are accessible and usable across various devices, from desktop computers to smartphones and tablets.
5.3 Accessibility: Tailwind CSS promotes accessibility by enabling developers to implement accessibility features directly through utility classes. This includes ensuring proper color contrast, managing focus states, and optimizing text readability, crucial for making library resources accessible to all users.
5.4 Performance: Tailwind CSS optimizes website performance by generating minimal and efficient CSS code. Unlike traditional frameworks that may include unused styles, Tailwind CSS generates only the necessary styles based on the utility classes used in the markup, leading to faster load times.
5.5 Consistency: Libraries often have numerous webpages and sections. Tailwind CSS promotes design consistency by facilitating the creation of reusable components and consistent styles across different parts of the website. This ensures that users have a cohesive experience as they navigate through various library resources.
5.6 Rapid Prototyping: Tailwind CSS accelerates the prototyping phase of website development by allowing developers to quickly mock up designs using utility classes. This speeds up the iteration process, enabling libraries to efficiently refine their website's layout and functionality based on user feedback and testing.
5.7 Maintenance: With Tailwind CSS, maintenance becomes more manageable as changes to the design or layout can be implemented swiftly through adjustments to the utility classes. This reduces the risk of introducing

unintended side effects or breaking existing functionality, crucial for maintaining a reliable library website.

5.8 Scalability: Tailwind CSS is highly scalable, making it suitable for large-scale library websites that may need to handle extensive amounts of content and diverse user interactions. The framework's modular approach allows developers to add new features or sections to the website without compromising performance or design consistency.

5.9 Community and Support: Tailwind CSS has a vibrant community and extensive documentation that provide valuable resources, tutorials, and plugins. This support network ensures that libraries adopting Tailwind CSS can access help and guidance when implementing new features or troubleshooting issues.

5.10 Future-Proofing: By adopting Tailwind CSS, libraries future-proof their websites against evolving design trends and technological advancements. The framework's focus on utility-first design principles ensures that websites remain adaptable and can easily incorporate new functionalities or design enhancements as needed.

5.11 Rapid Development: Speed up your development workflow with pre-built utility classes.

5.12 Flexibility: Customize styles without leaving your HTML file, offering unparalleled flexibility.

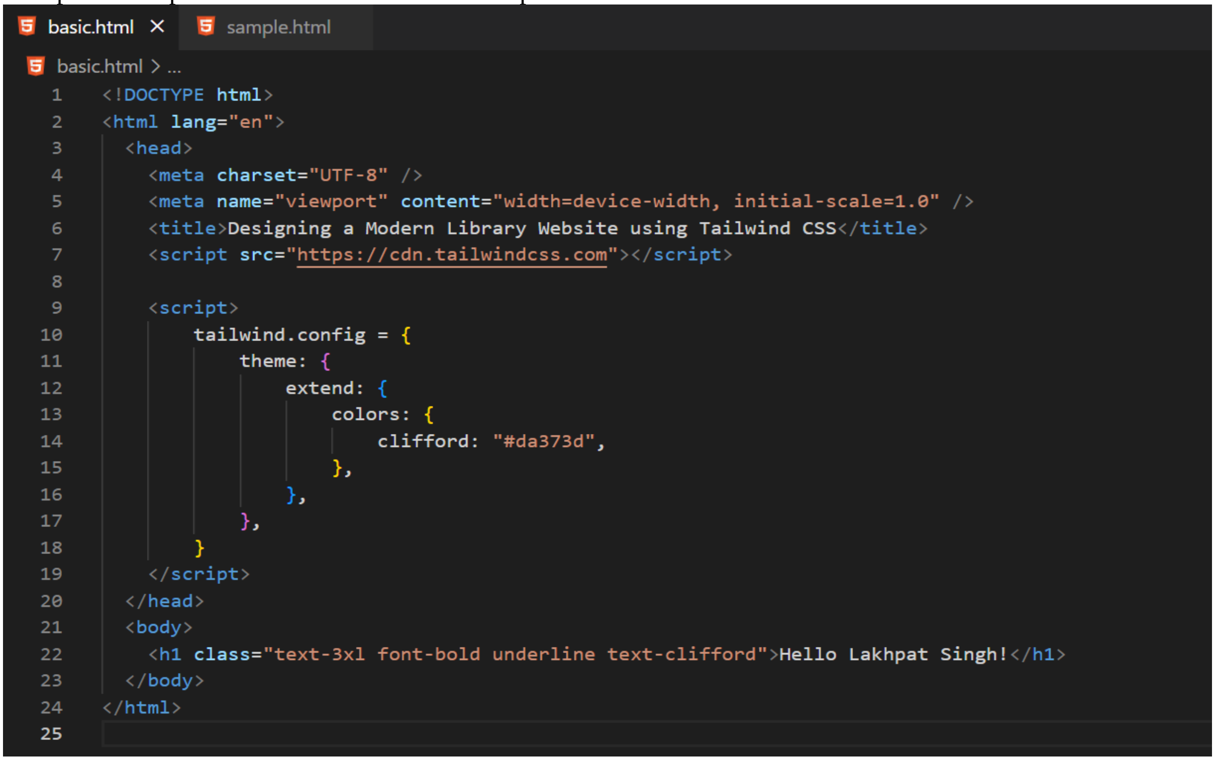
6.0 Tailwind Integration in Website:

Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file.

It's fast, flexible, and reliable — with zero-runtime. Here are the two options by which we can use the tailwind CSS. These are:-

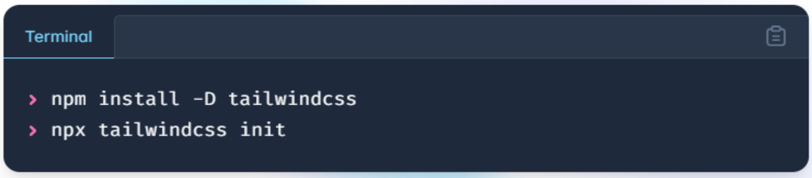
Option 1: Using CDN (Content Delivery Network)

Add the following line in the '<head>' section of your HTML file to include Tailwind's CSS.
<script src="https://cdn.tailwindcss.com"></script>

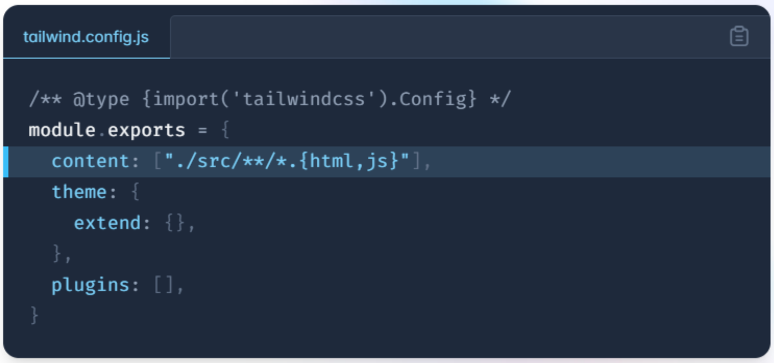


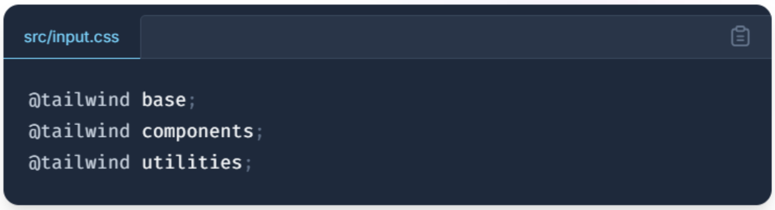
```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Designing a Modern Library Website using Tailwind CSS</title>
    <script src="https://cdn.tailwindcss.com"></script>

    <script>
        tailwind.config = {
            theme: {
                extend: {
                    colors: {
                        clifford: "#da373d",
                    },
                },
            },
        }
    </script>
  </head>
  <body>
    <h1 class="text-3xl font-bold underline text-clifford">Hello Lakhpat Singh!</h1>
  </body>
</html>

```

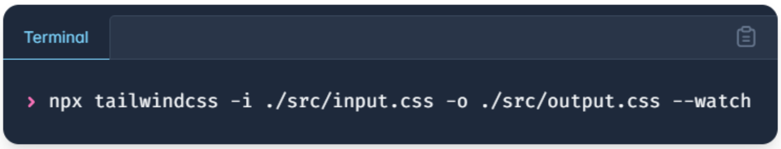
Option 2: Install Tailwind CSS

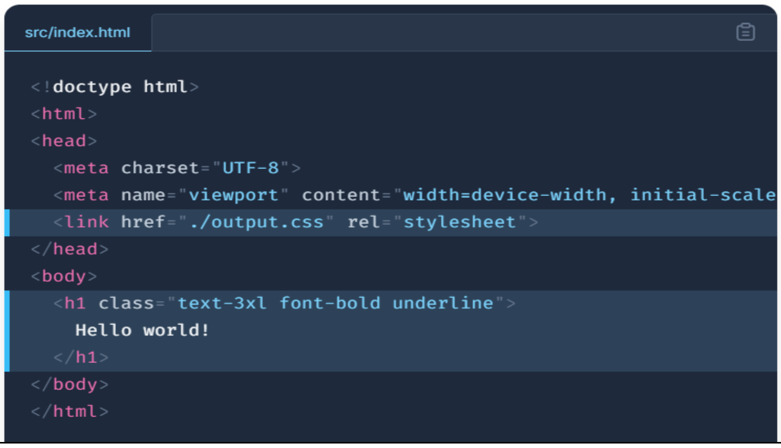
Install tailwindcss via npm, and create your tailwind.config.js file.	

```
› npm install -D tailwindcss
› npx tailwindcss init
```

<table>
<tr>
<td>Configure your template paths
Add the paths to all of your template files in your tailwind.config.js file.</td>
<td>


```
tailwind.config.js

/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{html,js}"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```
</td>
</tr>
<tr>
<td>Add the Tailwind directives to your CSS
Add the @tailwind directives for each of Tailwind’s layers to your main CSS file.</td>
<td>


```
src/input.css

@tailwind base;
@tailwind components;
@tailwind utilities;
```
</td>
</tr>
<tr>
<td>Start the Tailwind CLI build process
Run the CLI tool to scan your template files for classes and build your CSS.</td>
<td>


```
Terminal

> npx tailwindcss -i ./src/input.css -o ./src/output.css --watch
```
</td>
</tr>
<tr>
<td>Start using Tailwind in your HTML
Add your compiled CSS file to the <head> and start using Tailwind’s utility classes to style your content.</td>
<td>


```
src/index.html

<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale
  <link href="./output.css" rel="stylesheet">
</head>
<body>
  <h1 class="text-3xl font-bold underline">
    Hello world!
  </h1>
</body>
</html>
```
</td>
</tr>
</table>

7.0 Designing a Modern Library Website using Tailwind:

Designing a library website using Tailwind CSS can be an efficient and effective way to create a clean, responsive, and modern website. Below is a step-by-step guide to help you get started with designing a library website, including example code snippets and Tailwind CSS classes.

7.1 Set up Your Project

To get started, you’ll need to set up Tailwind CSS in your project. You can do this using a variety of methods, but the easiest way is to use a CDN (Content delivery Network) for quick prototyping. Or Install Tailwind CSS in local environment as mentioned above point.

7.2 Basic Structure

- Header: Includes navigation and library logo.
- Hero Section: Features a welcome message or a call-to-action.
- Search Bar: To allow users to search for books.
- Featured Books: Displays some of the library's featured books.
- Library Sections: Information about different library sections.
- Library Services: Information about different library services.
- Footer: Contact information, social media links, etc.

7.3 Header
Create a navigation bar with a logo and menu items:

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Designing a Modern Library Website using Tailwind CSS</title>
  <script src="https://cdn.tailwindcss.com"></script>
 </head>
 <body>
  <header class="bg-blue-800 text-white p-4">
   <div class="container mx-auto flex justify-between items-center">
    <div class="text-xl font-bold">Main Library</div>
    <nav class="space-x-4">
     <a href="#" class="hover:text-gray-300">Home</a>
     <a href="#search" class="hover:text-gray-300">Search</a>
     <a href="#featured" class="hover:text-gray-300">Featured Books</a>
     <a href="#sections" class="hover:text-gray-300">Sections</a>
     <a href="#services" class="hover:text-gray-300">Services </a>
     <a href="#contact" class="hover:text-gray-300">Contact</a>
    </nav>
    <div class="text-xl font-bold">Login</div>
   </div>
  </header>
  <div class="container mx-20 p-4">
       <h1 class="text-3xl font-bold underline text-clifford">
     Welcome to Main Library of Lakhpat Singh
       </h1>
  </div>
 </body>
</html>
```

7.4 Hero Section
Add a hero section with a welcome message and a call-to-action button:

```
  <section class="bg-cover bg-center h-screen" style="background-image: url('library-hero.jpg');">
    <div class="bg-black bg-opacity-50 h-full flex items-center justify-center">
      <div class="text-center text-white px-6">
        <h1 class="text-4xl md:text-6xl font-bold mb-4">Welcome to Main Library</h1>
        <p class="text-lg md:text-2xl mb-6">Explore our collection of books and resources.</p>
        <a href="#search" class="bg-yellow-500 text-black py-2 px-4 rounded hover:bg-yellow-
```

```
         600">Find a Book
         </a>
       </div>
     </div>
   </section>
```

7.5 Search Bar
Add a search bar for finding books:

```
  <section id="search" class="py-8 bg-gray-100">
     <div class="container mx-auto text-center">
       <h2 class="text-2xl font-semibold mb-4">Search for a Book (Web-OPAC)</h2>
       <input type="text" placeholder="Enter book title, author, or subject" class="p-2 w-full max-w-lg border
border-gray-300 rounded">
     </div>
   </section>
```

7.6 Featured Books
Showcase some featured books:

```
   <section id="featured" class="py-8">
    <div class="container mx-auto text-center">
     <h2 class="text-2xl font-semibold mb-6">Featured Books</h2>
     <div class="grid grid-cols-1 md:grid-cols-3 gap-6">
      <div class="border border-gray-200 p-4 rounded">
       <img src="book1.jpg" alt="Book 1" class="w-full h-40 object-cover mb-4 rounded" />
       <h3 class="text-lg font-semibold mb-2">Book Title 1</h3>
       <p class="text-gray-600">Author 1</p>
      </div>
      <div class="border border-gray-200 p-4 rounded">
       <img src="book2.jpg" alt="Book 2" class="w-full h-40 object-cover mb-4 rounded" />
       <h3 class="text-lg font-semibold mb-2">Book Title 2</h3>
       <p class="text-gray-600">Author 2</p>
      </div>
      <div class="border border-gray-200 p-4 rounded">
       <img src="book3.jpg" alt="Book 3" class="w-full h-40 object-cover mb-4 rounded" />
       <h3 class="text-lg font-semibold mb-2">Book Title 3</h3>
       <p class="text-gray-600">Author 3</p>
      </div>
     </div>
    </div>
   </section>
```

7.7 Library Sections

Describe different sections of the library:

```
<section id="sections" class="py-8 bg-gray-100">
  <div class="container mx-auto text-center">
    <h2 class="text-2xl font-semibold mb-6">Library Sections</h2>
    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">Fiction</h3>
        <p class="text-gray-600">
          Explore a wide variety of fiction books including classics, contemporary novels, and more.
        </p>
      </div>
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">Non-Fiction</h3>
        <p class="text-gray-600">Discover non-fiction books on history, science, self-help, and more.</p>
      </div>
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">Children’s Books</h3>
        <p class="text-gray-600">Find children’s books from picture books to young adult novels.</p>
      </div>
    </div>
  </div>
</section>
```

7.8 Library Services

Describe different sections of the library:

```
<section id="services" class="py-8 bg-blue-100">
  <div class="container mx-auto text-center">
    <h2 class="text-2xl font-semibold mb-6">Library Services</h2>
    <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6">
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">Circulation Service</h3>
        <p class="text-gray-600">Here we can issue or return the library documents as required.</p>
      </div>
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">Reference Service</h3>
        <p class="text-gray-600">Discover Reference books on history, science, and more...</p>
      </div>
      <div class="p-4 border border-gray-200 rounded">
        <h3 class="text-xl font-semibold mb-2">OPAC Service</h3>
        <p class="text-gray-600">Find the library books with Author, Title, Subject and more...</p>
      </div>
    </div>
  </div>
</section>
```

7.9 Footer

Add a footer with contact information and social media links:

```
  <footer id="contact" class="bg-blue-800 text-white py-4">
    <div class="container mx-auto text-center">
      <p class="mb-2">&copy; 2024 Main Library of Lakhpat Singh. All rights reserved.</p>
      <p class="mb-4">Jagadhri, Yamunanagar, 135003</p>
      <div class="space-x-4">
        <a href="#" class="hover:text-gray-300">Facebook</a>
        <a href="#" class="hover:text-gray-300">Twitter</a>
        <a href="#" class="hover:text-gray-300">Instagram</a>
      </div>
    </div>
  </footer>
```

7.10 Additional Features

Depending on your needs, you can add more features:
- Events Calendar: Use a plugin or a custom solution for event management.
- Online Catalogue: If you have a digital catalogue, consider integrating or linking to it.
- User Login/Account Management: For advanced features, you might want to add user authentication.

7.11 Responsive Design

Tailwind CSS provides a range of utilities for responsive design. Ensure your layout looks good on all screen sizes by testing and adjusting classes for different breakpoints.

```
  <div class="p-4 md:p-6 lg:p-8">
    <!-- Content here -->
  </div>
```

7.12 Tailwind CSS Utilities Reference

Here are some useful Tailwind utilities for web design:
- Spacing: ‘p-4’, ‘m-4’, ‘pt-4’, ‘pb-4’, etc.
- Typography: ‘text-xl’, ‘font-bold’, ‘text-gray-600’, etc.
- Layout: ‘flex’, ‘grid’, ‘container’, ‘mx-auto’, etc.
- Backgrounds: ‘bg-gray-100’, ‘bg-cover’, ‘bg-center’, etc.
- Borders: ‘border’, ‘border-gray-200’, ‘rounded’, etc.

7.13 Quick Reference Table

Website	URL	Key Features
Tailwind Components	tailwindcomponents.com	Large collection of components and templates.
Tailwind UI	tailwindui.com	Free components from a paid kit.
Tailwind Toolbox	tailwindtoolbox.com	Components, templates, and free resources.
DaisyUI	daisyui.com	Free components and themes for Tailwind.
Flowbite	flowbite.com	Free components and templates.
Tailwind Starter Kit	creativetim.com	Free components and example pages.
Windster	windster.com	Free components and templates.
Components.dev	components.dev	Community-driven components library.
Tailwind Kit	tailwindkit.com	Free components and templates.
Notus Tailwind	creativetim.com	Free Tailwind CSS kit with components and templates.

Feel free to explore these resources and choose the one that best fits your project's needs!

8.0 Downfalls of the Tailwind CSS

As futuristic and revolutionary tailwind CSS is, it does have some flaws.

8.1 Repeating classes: -

While the idea of having utility classes is really great, there are some issues with repeating these classes. For example:-

```
  <h1 class="bold text-2xl py-2 text-blue">Title One</h1>
  <h1 class="bold text-2xl py-2 text-red">Title Two</h1>
  <h1 class="bold text-2xl py-2 text-green">Title Three</h1>
```

While working with CSS, I can write a common class for repetitive styles and add that same class to markup.

8.2 Readability:-

It is the biggest downfall of tailwind CSS, readability. While adding these classes, the readability can be challenging.

Suppose you have a button like:-

```
  <button class="bg-red-500 text-white font-bold py-2 px-4 rounded mt-5 hover:bg-red-400 border border-
red">
    Read More
  </button>
```

8.3 Learning Curve and Complexity:

Tailwind CSS operates on a utility-first approach, which requires developers to learn and effectively use its extensive set of utility classes. This paradigm shift from traditional CSS frameworks may steepen the learning curve for new team members and demand additional time for proficiency development.

8.4 File Size and Performance Concerns:

Tailwind CSS generates a large stylesheet due to its comprehensive utility classes. If not managed properly, this can lead to bloated CSS files and slower load times, particularly detrimental for users with limited internet bandwidth or accessing the website on mobile devices.

8.5 Dependency on Configuration and Tooling:

Tailwind CSS heavily relies on configuration files and build tooling for customization and optimization. Libraries may need to invest additional effort in setting up and fine-tuning these configurations to achieve the desired design aesthetics and performance outcomes.

9.0 Conclusion:

As discussed above, tailwind CSS comes with some downfalls, but the advantages it offers are far bigger than these tradeoffs. The utility-first approach of the tailwind CSS makes it a great framework choice for design-oriented developers. In the end, choosing tailwind CSS for your next project can be a worthy inclusion.

Designing a modern library website using Tailwind CSS presents a compelling case for leveraging contemporary web design principles to enhance user experience and functionality. Tailwind CSS's utility-first approach facilitates rapid prototyping and streamlined development, crucial for maintaining a responsive and accessible interface across various devices. By utilizing its extensive set of pre-defined utility classes, developers can efficiently implement intricate design components and responsive layouts without the constraints of traditional CSS frameworks.

Moreover, Tailwind CSS promotes design consistency and scalability through its customizable configuration options, allowing libraries to tailor their website's aesthetics and functionality to meet specific user needs effectively. This flexibility is particularly advantageous for libraries aiming to provide intuitive navigation, seamless search functionalities, and interactive features that enhance the overall browsing experience for patrons.

In conclusion, adopting Tailwind CSS for designing a modern library website not only enhances visual appeal but also improves usability and accessibility. It empowers libraries to create a digital space that fosters engagement, facilitates information discovery, and reinforces their role as a vital resource in the community's digital landscape.

10.0 References:

i. https://www.geeksforgeeks.org/getting-started-with-tailwind-css/
ii. https://tailwindcss.com
iii. https://tailwindcss.com/docs/installation
iv. https://en.wikipedia.org/wiki/Tailwind_CSS
v. https://www.material-tailwind.com/blog/7-reasons-why-you-should-use-tailwind-css
vi. https://www.geeksforgeeks.org/tailwind-css-vs-bootstrap/
vii. https://adamwathan.me/going-full-time-on-tailwind-css/
viii. https://xenox.dev/tailwind-css-utility-first-css-framework/