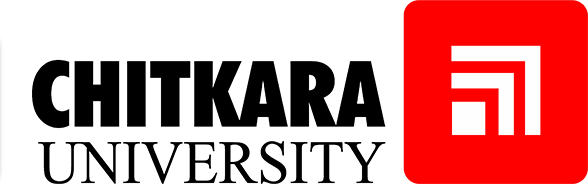
**Front End Engineering-II**

Project Report

Semester-IV (Batch-2022)

**IMAGE SEARCH APP**



**Supervised By: Submitted By:**

Dr. Raveesh Samkaria Yazvi Balooni

2210990747

(G-14)

**Department of Computer Science and Engineering**

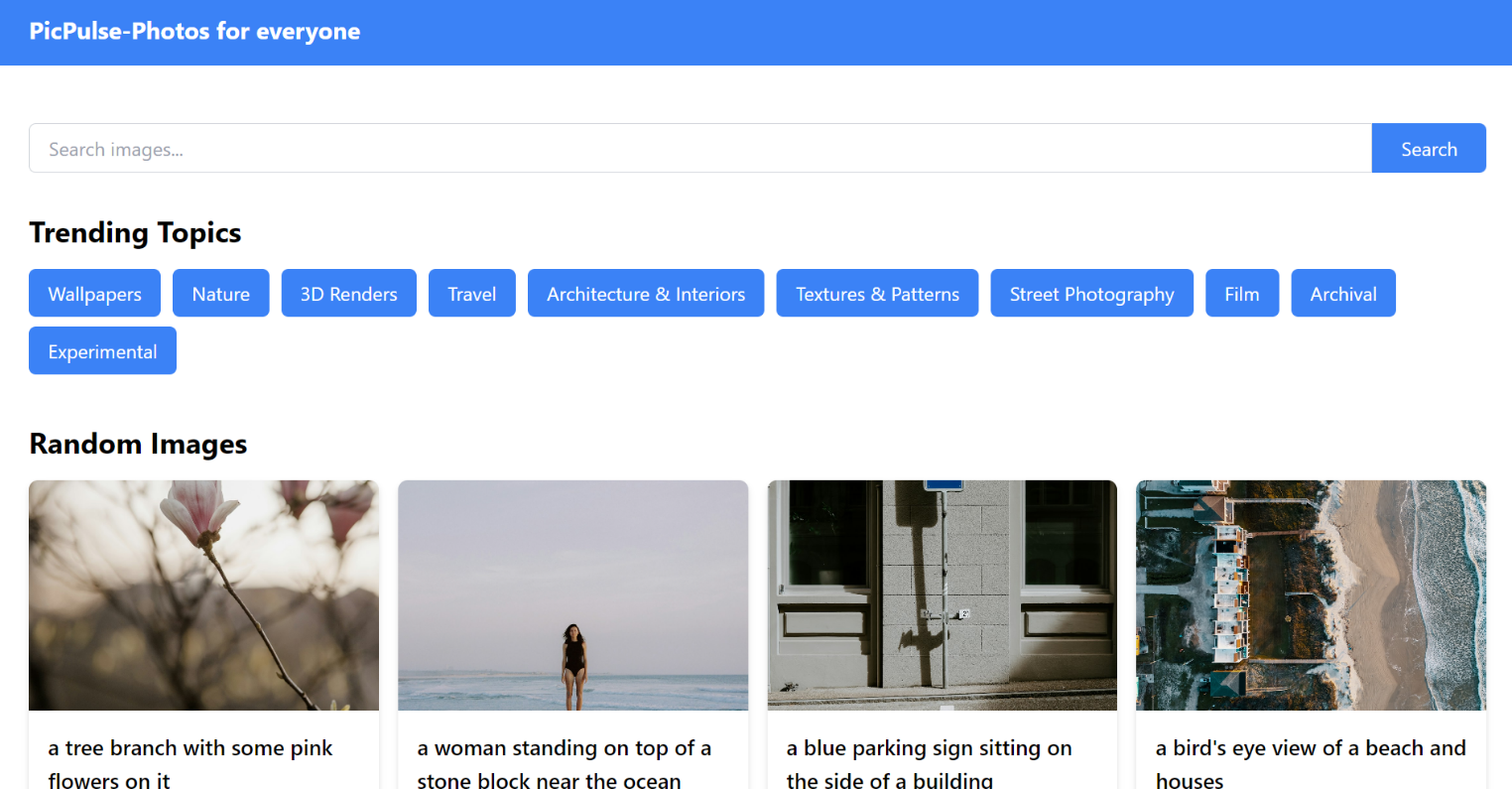
**Chitkara University Institute of Engineering &Technology,**

**Chitkara University, Punjab**

# Abstract

Crafting the foundational structure, this image search app's layout and elements guide users seamlessly through its platform. Tailwind CSS adds an artistic touch, meticulously fashioning the visual presentation with elegant styles and responsive design principles. JavaScript breathes life into the interface, enabling dynamic interactions and enhancing user experience with features like live search and voice input. Bootstrap, the versatile framework, underpins the app, ensuring compatibility across devices and streamlining development with its robust set of components and utilities. Set against a backdrop reminiscent of artistic inspiration, the search bar stands as a beacon, offering a gateway to endless visual discoveries.

Each image, meticulously curated and displayed with precision, embodies the app's ethos—simplicity in complexity. Through intuitive controls and responsive elements, users effortlessly navigate, while subtle hover effects add a touch of interactivity. With every search query, a symphony of colors and textures unfolds, painting a vivid tapestry of inspiration.



# INDEX

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Title** | **Page Number(s)** |
| 1 | Introduction | 4 |
| 2 | Problem Statement | 4 |
| 3 | Software Requirements | 5,6 |
| 4 | Proposed Design | 7-14 |
| 5 | Results | 15-18 |
| 6 | References | 19 |

# 1.Introduction

In today's digital world, the ability to search for and discover images has become an essential part of our online experience. From finding inspiration for creative projects to identifying objects and places, image search apps play a crucial role in how we interact with visual content. This project aims to create an innovative image search app that combines powerful search functionality with a sleek and intuitive user interface, providing users with a seamless and enjoyable image search experience.

**1.1 Background:**

Image search technology has advanced significantly in recent years, thanks to advancements in machine learning and computer vision. These technologies enable image search apps to analyse and understand the content of images, making it easier for users to find exactly what they're looking for. By leveraging these technologies, our app will be able to provide accurate and relevant search results, helping users discover new and interesting visual content.

**1.2 Objectives:**

The primary objective of this project is to develop an image search app that offers a fast, accurate, and user-friendly search experience. The app will leverage advanced image recognition algorithms to analyse and categorize images, enabling users to search for images using keywords, tags, or even by uploading an image. Additionally, the app will prioritize responsive design, ensuring that it works seamlessly across a wide range of devices and screen sizes.

**1.3 Significance:**

This project is significant as it demonstrates the power of modern image search technologies and their impact on user experiences. By developing an image search app that is both powerful and easy to use, we hope to showcase the potential of image search technology in enhancing our online interactions with visual content.

# Problem Statement

While there are many image search apps available today, not all of them offer a truly seamless and intuitive user experience. Some apps may have limited search capabilities or may not provide accurate search results. This project aims to address these issues by developing an image search app that is both powerful and easy to use, providing users with a fast and reliable way to search for images.

# Software Requirements

**I. Integrated Development Environment (IDE):**

• Integrated Development Environment (IDE): Visual Studio Code (VS Code) will be used as the primary code editor for developing the image search app, thanks to its robust features and extensions ecosystem.

1. **Technology Requirements:** 
   * HTML5 (Hypertext Markup Language) will be used to structure the web application, creating elements for the search bar, results display, and user interface components.
   * CSS3 (Cascading Style Sheets) will be used for styling the application, ensuring a visually appealing and coherent design across different components and screen sizes.
   * JavaScript (ES6+) will be the programming language of choice for implementing interactive features, dynamic updates, time calculations, and handling user interactions within the application.

1. **Utility-first CSS Framework:**

* Tailwind CSS will be utilized as the primary CSS framework, offering high customizability and a utility-first approach to styling. This will streamline the design process, allowing for a modern and responsive design with minimal effort.

**IV. Version Control and Collaboration:**

* Git, a distributed version control system, will be utilized for tracking changes in the project codebase, enabling efficient collaboration, and facilitating code reviews and merges.
* The project repository will be hosted on GitHub, a popular platform for code hosting, collaboration, and issue tracking.

**V.**  **Browser Compatibility and Performance:**

* The application will be developed with a focus on cross-browser compatibility, ensuring it functions correctly and consistently across the latest versions of popular web browsers. Performance optimization techniques will be implemented to ensure smooth and efficient rendering, minimizing resource consumption and providing an optimal user experience.

# Proposed Design

**User Interface:**

* The application will feature a responsive and visually appealing layout using Bootstrap 5, ensuring a consistent experience across various devices and screen sizes.
* A card-based design will be employed to organize the different components of the application, such as the digital clock, analog clock, and time format/date display options, in an intuitive and user-friendly manner.

**Development Technologies:**

* HTML5 semantic elements will be utilized for structuring the application, ensuring proper semantics and accessibility.
* CSS will be responsible for styling the application, creating a visually appealing and cohesive design.
* JavaScript will be used for implementing dynamic UI updates, handling user interactions, and managing the clock logic and time calculations.

**User Experience:**

* Real-time feedback and interactive elements, such as smooth animations and transitions, will be implemented to create an engaging and responsive user interface.
* Cross-browser compatibility will be a priority, ensuring the application functions consistently across the latest versions of popular web browsers.
* Responsiveness will be a key focus, with the application designed to adapt seamlessly to different devices and screen sizes, providing an optimal user experience on both desktop and mobile platforms.

**Testing and Quality Assurance:**

* Comprehensive testing, including manual and unit tests, will be conducted to ensure the application's functionality and user interface consistency.

**Documentation and Deployment:**

* Detailed documentation, including installation instructions, usage guidelines, and relevant information for developers and users, will be provided.
* The application will be deployed on a web server with a domain, ensuring accessibility and ease of use for end-users.

**Dynamic Background:**

* The background image of the application will change dynamically based on the current time, creating a visually appealing and immersive experience for the user.
* Different background images will be associated with specific time periods, such as morning, afternoon, evening, and night, to create a seamless and cohesive design.

* 1. **File Structure**

Establishing a meticulous organization of files and folders to uphold uniform file paths and a clutter-free arrangement.



* 1. **HTML Code Structure**

The provided screenshots showcase the HTML code for our Digital and Analog Clock project, depicting the layout and content of our web pages in code format.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>PicPulse - Photos for Everyone</title>

    <link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.19/dist/tailwind.min.css" rel="stylesheet">

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <nav class="bg-blue-500 text-white px-6 py-4 w-full mb-6">

        <div class="container mx-auto flex justify-between items-center">

            <h1 class="text-xl font-bold">PicPulse-Photos for everyone</h1>

        </div>

    </nav>

    <div class="container mx-auto p-6">

        <div id="searchContainer">

            <div class="flex mb-2">

                <input type="text" id="searchInput"

                    class="flex-1 px-4 py-2 border border-gray-300 rounded-l-md focus:outline-none focus:border-blue-500"

                    placeholder="Search images...">

                <button id="searchButton"

                    class="px-6 py-2 bg-blue-500 text-white rounded-r-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600">Search</button>

            </div>

            <button id="backButton"

                class="px-6 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600 hidden">Back to Trending</button>

            <div id="trendingTopics" class="mt-8">

                <h2 class="text-2xl font-bold mb-4">Trending Topics</h2>

                <div class="flex flex-wrap">

                    <ul id="trendingTopicsList" class="flex flex-wrap">

                    </ul>

                </div>

            </div>

            <!-- Random Images Section -->

            <div class="mt-8 hidden" id="randomImagesContainer">

                <h2 class="text-2xl font-bold mb-4">Random Images</h2>

                <div id="randomImages"

                    class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4"></div>

            </div>

        </div>

        <!-- Search Results Section -->

        <div id="searchResults"

            class="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-4 mt-8"></div>

        <!-- Image Preview Modal -->

        <div id="imagePreviewModal"

            class="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center hidden z-50">

            <div class="bg-white rounded-lg p-4 max-w-lg max-h-full overflow-auto">

                <img id="previewImage" src="" alt="Preview Image">

                <button id="closeModalBtn"

                    class="mt-4 block px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600">Close</button>

            </div>

        </div>

    </div>

    <script src="yazvi.js"></script>

</body>

</html>

* 1. **CSS Code Structure**

The screenshots exhibit the CSS code for our Digital and Analog Clock project, demonstrating the styling and visual presentation of our web pages in code format.

.trending-topic-btn {

  display: inline-block;

  margin-right: 10px;

}

* 1. **Javascript Code Structure**

The screenshots depict the JavaScript code for our Digital and Analog Clock project, showcasing the interactive features and dynamic functionalities of our web pages in code format.

const accessKey = 'mkJQzmgYL2zHgdrXILvewx4G-anZzqDSib\_U3yZ06Ro';

        const apiUrl = 'https://api.unsplash.com/';

        const perPage = 12;

        const searchContainer = document.getElementById('searchContainer');

        const searchInput = document.getElementById('searchInput');

        const searchButton = document.getElementById('searchButton');

        const searchResults = document.getElementById('searchResults');

        const trendingTopics = document.getElementById('trendingTopics');

        const trendingTopicsList = document.getElementById('trendingTopicsList');

        const randomImagesContainer = document.getElementById('randomImagesContainer');

        const randomImages = document.getElementById('randomImages');

        const previewModal = document.getElementById('imagePreviewModal');

        const previewImage = document.getElementById('previewImage');

        const closeModalBtn = document.getElementById('closeModalBtn');

        const backButton = document.getElementById('backButton');

        searchButton.addEventListener('click', () => {

            searchImages();

        });

        searchInput.addEventListener('keydown', (event) => {

            if (event.key === 'Enter') {

                searchImages();

            }

        });

        fetchTrendingTopics();

        fetchRandomImages();

        async function fetchRandomImages() {

            try {

                const response = await fetch(`${apiUrl}photos/random?count=${perPage}&client\_id=${accessKey}`);

                if (!response.ok) {

                    throw new Error('Failed to fetch random images');

                }

                const data = await response.json();

                if (data.length > 0) {

                    randomImagesContainer.classList.remove('hidden');

                }

                randomImages.innerHTML = '';

                data.forEach(result => {

                    const imageCard = `

                    <div class="image-card bg-white rounded-lg shadow-md overflow-hidden">

                        <img src="${result.urls.regular}" alt="${result.alt\_description || 'Unsplash Image'}"

                            class="w-full h-48 object-cover cursor-pointer">

                        <div class="p-4">

                            <h2 class="text-lg font-semibold mb-2">${result.alt\_description || 'Image'}</h2>

                            <p class="text-gray-600">Photo by ${result.user.name}</p>

                            <a href="${result.links.download}" target="\_blank"

                                class="mt-4 block px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600">Download</a>

                        </div>

                    </div>

                `;

                    randomImages.insertAdjacentHTML('beforeend', imageCard);

                });

                // Add event listeners for image click and close modal button

                document.querySelectorAll('.image-card img').forEach(img => {

                    img.addEventListener('click', () => openPreviewModal(img.src));

                });

                closeModalBtn.addEventListener('click', closePreviewModal);

            } catch (error) {

                console.error('Error fetching random images:', error.message);

            }

        }

        async function fetchTrendingTopics() {

            try {

                const response = await fetch(`${apiUrl}topics?client\_id=${accessKey}`);

                if (!response.ok) {

                    throw new Error('Failed to fetch trending topics');

                }

                const data = await response.json();

                trendingTopicsList.innerHTML = '';

                data.forEach(topic => {

                    const topicItem = document.createElement('li');

                    topicItem.classList.add('mb-2');

                    topicItem.innerHTML = `

                        <button class="trending-topic-btn bg-blue-500 text-white px-4 py-2 rounded-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600">${topic.title}</button>

                    `;

                    topicItem.querySelector('button').addEventListener('click', () => {

                        searchInput.value = topic.title;

                        searchImages();

                    });

                    trendingTopicsList.appendChild(topicItem);

                });

            } catch (error) {

                console.error('Error fetching trending topics:', error.message);

            }

        }

        async function searchImages() {

            const searchTerm = searchInput.value.trim();

            if (searchTerm === '') {

                // If there's no search term, show the trending topics and random images

                trendingTopics.style.display = 'block';

                randomImagesContainer.style.display = 'block';

                backButton.style.display = 'none'; // Hide back button

                fetchTrendingTopics();

                fetchRandomImages();

                return;

            }

            // Hide the trending topics and random images when there's a search term

            trendingTopics.style.display = 'none';

            randomImagesContainer.style.display = 'none';

            backButton.style.display = 'block'; // Show back button

            try {

                const response = await fetch(`${apiUrl}search/photos?query=${searchTerm}&per\_page=${perPage}&client\_id=${accessKey}`);

                if (!response.ok) {

                    throw new Error('Failed to search images');

                }

                const data = await response.json();

                searchResults.innerHTML = '';

                data.results.forEach(result => {

                    const imageCard = `

                        <div class="image-card bg-white rounded-lg shadow-md overflow-hidden">

                            <img src="${result.urls.regular}" alt="${result.alt\_description || 'Unsplash Image'}"

                                class="w-full h-48 object-cover cursor-pointer">

                            <div class="p-4">

                                <h2 class="text-lg font-semibold mb-2">${result.alt\_description || 'Image'}</h2>

                                <p class="text-gray-600">Photo by ${result.user.name}</p>

                                <a href="${result.links.download}" target="\_blank"

                                    class="mt-4 block px-4 py-2 bg-blue-500 text-white rounded-md hover:bg-blue-600 focus:outline-none focus:bg-blue-600">Download</a>

                            </div>

                        </div>

                    `;

                    searchResults.insertAdjacentHTML('beforeend', imageCard);

                });

                document.querySelectorAll('.image-card img').forEach(img => {

                    img.addEventListener('click', () => openPreviewModal(img.src));

                });

                closeModalBtn.addEventListener('click', closePreviewModal);

            } catch (error) {

                console.error('Error searching images:', error.message);

            }

        }

        function openPreviewModal(imageUrl) {

            previewModal.classList.remove('hidden');

            previewImage.style.width = 'auto';

            previewImage.style.height = 'auto';

            previewImage.src = imageUrl;

            previewImage.onload = function() {

                const viewportHeight = window.innerHeight;

                const viewportWidth = window.innerWidth;

                const imageWidth = previewImage.width;

                const imageHeight = previewImage.height;

                const aspectRatio = imageWidth / imageHeight;

                const maxHeight = viewportHeight \* 0.8;

                const maxWidth = viewportWidth \* 0.8;

                if (aspectRatio > 1) {

                    // Landscape or square image

                    if (imageWidth > maxWidth) {

                        previewImage.style.width = `${maxWidth}px`;

                        previewImage.style.height = 'auto';

                    }

                } else {

                    // Portrait image

                    if (imageHeight > maxHeight) {

                        previewImage.style.height = `${maxHeight}px`;

                        previewImage.style.width = 'auto';

                    }

                }

            };

        }

        function closePreviewModal() {

            previewModal.classList.add('hidden');

        }

        // Add event listener to the back button

        backButton.addEventListener('click', () => {

            // Show the trending topics and random images

            trendingTopics.style.display = 'block';

            randomImagesContainer.style.display = 'block';

            // Hide the search results section

            searchResults.innerHTML = '';

            // Hide the back button

            backButton.style.display = 'none';

            // Clear the search input

            searchInput.value = '';

            // Fetch trending topics

            fetchTrendingTopics();

            // Fetch random images

            fetchRandomImages();

        });

# Results

The displayed results showcase the synchronized digital and analog clocks, alongside dynamically changing background images and greeting messages based on the current time. Through rigorous testing and user feedback, the following key outcomes have been achieved: **Git Hub Repository Link:** <https://github.com/itsrudra143/Digital-and-Analog-Clock>

**Git Hub Pages Link**:<https://itsrudra143.github.io/Digital-and-Analog-Clock/>

These screenshots capture different instances of my project, showcasing its appearance and functionality at various points in time.



## Morning Ambiance



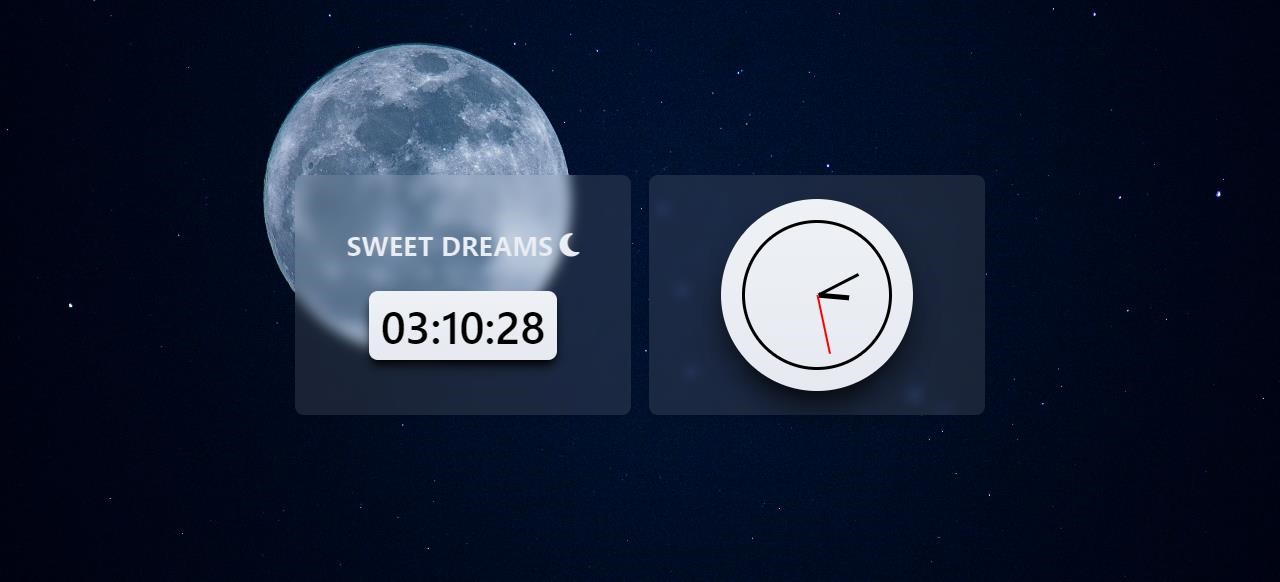
## Afternoon Serenity



## Evening Tranquility



**Nighttime Stillness**



**Midnight Mystery**

Our project focuses on digital and analog clocks, presenting accurate time

Utilizing HTML, CSS, Tailwind, and JavaScript, we've crafted a seamless and visually appealing time-telling interface.

JavaScript ensures synchronization between the digital and analog clocks, offering precise timekeeping.

A standout feature of our project is its adaptability—the background image changes dynamically to reflect the time of day.

Whether it's the serene glow of morning, the warm hues of afternoon, the calming twilight of evening, the tranquil darkness of night, or the enigmatic ambiance of midnight, our project adjusts the background image accordingly.

Through this integration of technology and design, our project offers both functionality and aesthetic appeal, enhancing the user experience with each passing hour.

# References

* **Mozilla Developer Network (MDN) - HTML, CSS, JS Documentation:**

Website: https://developer.mozilla.org/

Description: MDN offers comprehensive documentation on HTML, CSS, and JavaScript, covering everything from basic syntax to advanced concepts and APIs.

* **W3Schools - HTML, CSS, JavaScript Tutorials:**

Website: https://www.w3schools.com/

Description: W3Schools provides beginner-friendly tutorials and references for HTML, CSS, and JavaScript, along with interactive code examples.

* **Tailwind CSS Documentation:**

Website: https://tailwindcss.com/docs

Description: The official Tailwind CSS documentation provides detailed guidance on using Tailwind CSS for building modern and responsive web interfaces. It includes utility classes, customization options, and best practices.

* **CSS-Tricks:**

Website: https://css-tricks.com/

Description: CSS-Tricks is a web design community that offers articles, tutorials, and resources related to CSS, HTML, and frontend development. It covers a wide range of topics, including layout techniques, CSS animations, and frontend frameworks.

* **Smashing Magazine:**

Website**:** https://www.smashingmagazine.com/

Description: Smashing Magazine publishes articles, tutorials, and case studies on web design and development, including frontend technologies like HTML, CSS, JavaScript, and modern frameworks like Tailwind CSS. It also features interviews with industry experts and showcases of cutting-edge web projects.