Design Document: Portfolio Website

Overview

The portfolio is designed to showcase my developer skills, projects, and my experience in an interactive and engaging way. It is built with Next.js, a framework for React, due to its server-side rendering capabilities, route prefetching, and ease of deployment.

Technical Problem Statement

The main objective was to create a personal portfolio website that is fast, SEO-friendly, and easy to maintain. The technical challenge was to implement a modern web application architecture that facilitates the display of dynamic content, such as projects and experiences, with minimal load time.

Design Approach

The portfolio leverages Next.js's server-side rendering (SSR) for improved performance and SEO. The application's structure is modular, with components organized to encapsulate functionality and promote reusability.

Directory Structure

**pages**: Contains route-specific renderings, utilizing Next.js’s file-based routing system.

**public**: Houses static assets, ensuring that images and icons are served efficiently.

Src/app/components: Stores reusable UI components.

**styles**: Includes global styles and CSS configurations.

Component Separation

Components are separated into individual files within the src/app/components directory to encapsulate functionality and styles, allowing for better code management and reuse.

All the components are reusable and are used in Page.js to populate the homepage.

State Management

State within components is maintained when the data is tightly coupled with the UI logic, reducing props drilling and complexity. For instance, EmailSection.jsx contains its state as it manages form input internally.

Challenges and Solutions

Responsive design: Ensuring the portfolio is responsive across devices was a challenge that was addressed with Tailwind CSS for utility-first styles and responsive classes.

Email Form: It took me a lot of time to figure out the correct usage of SMTP API to send emails from visitors that land at my page and want to send me a message.

Rendering Tools

A hybrid approach was used, with **Server-Side Rendering (SSR)** for the initial load to improve SEO and loading times, and **Client-Side Rendering (CSR)** for dynamic user interactions to enhance the user experience.