

Coding Assistant - Next.js Tailwind Dark Mode

Create a layout and tailwind css files for a next.js SPA.
Please make the style modern, professional, and dark mode themed.

Here's a complete implementation for a modern, professional dark mode layout in a Next.js SPA using Tailwind CSS. This includes a layout component and a `globals.css` file to apply global Tailwind styles.

1. Install Tailwind CSS

First, set up Tailwind CSS in your Next.js project. If you haven't already:

```
bash

npx create-next-app@latest my-dark-mode-spa
cd my-dark-mode-spa
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init
```

2. Configure `tailwind.config.js`

Update the Tailwind configuration to enable dark mode and include necessary file paths:

```
javascript

/** @type {import('tailwindcss').Config} */
module.exports = {
  darkMode: 'class', // Enables manual dark mode toggling
  content: [
    './pages/**/*.{js,ts,jsx,tsx}',
    './components/**/*.{js,ts,jsx,tsx}',
  ],
  theme: {
    extend: {
      colors: {
        primary: '#1F2937', // Dark background
        accent: '#3B82F6', // Accent color (blue)
      }
    }
  }
}
```

```
        secondary: '#4B5563', // Secondary text
        light: '#F3F4F6', // Light text for contrast
    },
},
},
plugins: [],
};
```

3. globals.css

Create a global styles file for your project. Save this as `styles/globals.css`:

```
CSS

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

/* Custom Scrollbar Styling */
::-webkit-scrollbar {
    width: 8px;
}

::-webkit-scrollbar-thumb {
    background-color: #3B82F6; /* Accent color */
    border-radius: 4px;
}

body {
    @apply bg-primary text-light; /* Dark background with light text */
}
```

4. Layout Component (`components/Layout.js`)

This layout includes a responsive navbar and footer.

```

import { useState } from 'react';

export default function Layout({ children }) {
  const [darkMode, setDarkMode] = useState(true);

  const toggleDarkMode = () => {
    setDarkMode(!darkMode);
    document.documentElement.classList.toggle('dark', !darkMode);
  };

  return (
    <div className={`min-h-screen flex flex-col ${darkMode ? 'dark' : ''}`>
      {/* Navbar */}
      <nav className="bg-secondary p-4 text-light shadow-md">
        <div className="max-w-6xl mx-auto flex justify-between items-center">
          <h1 className="text-2xl font-bold">My Dark SPA</h1>
          <button
            onClick={toggleDarkMode}
            className="px-4 py-2 bg-accent rounded-md text-light hover:bg-opacity-
75"
          >
            Toggle {darkMode ? 'Light' : 'Dark'} Mode
          </button>
        </div>
      </nav>

      {/* Main Content */}
      <main className="flex-1 max-w-6xl mx-auto p-6">{children}</main>

      {/* Footer */}
      <footer className="bg-secondary p-4 text-center text-light">
        <p>© {new Date().getFullYear()} My Dark SPA. All Rights Reserved.</p>
      </footer>
    </div>
  );
}

```

5. Update `_app.js`

Use the `Layout` component in your application. Update `pages/_app.js`:

javascript

```
import '../styles/globals.css';
import Layout from '../components/Layout';

export default function MyApp({ Component, pageProps }) {
  return (
    <Layout>
      <Component {...pageProps} />
    </Layout>
  );
}
```

6. Example Home Page (`pages/index.js`)

Here's an example of a styled homepage:

javascript

```
export default function Home() {
  return (
    <div className="text-center">
      <h1 className="text-4xl font-extrabold mb-4">Welcome to My Dark SPA</h1>
      <p className="text-secondary">
        A modern, dark-mode themed single-page application built with Next.js and
        Tailwind CSS.
      </p>
    </div>
  );
}
```

Final Notes:

- The dark mode toggle manually switches between light and dark by toggling the `dark` class on the `<html>` element.
- You can customize the colors in `tailwind.config.js` for branding.
- Add more components (e.g., sidebars, cards) and apply the same principles for a consistent modern dark mode aesthetic.