Admin Dashboard: HTML & CSS Implementation

Aim

The aim of this lab is to create a responsive and functional administrative dashboard user interface using HTML and CSS. The dashboard will include a main content area and a collapsible sidebar, with a user-selectable theme (light and dark mode) to enhance user experience.

Objectives

1. **Structure the application:** Use semantic HTML tags to create a logical structure for the dashboard, including a header, a main content area, and a sidebar.
2. **Style the interface:** Apply CSS to style the layout, typography, and color schemes, ensuring the design is clean, modern, and easy to navigate.
3. **Implement theme switching:** Develop CSS variables and a JavaScript function to allow users to toggle between a light theme and a dark theme.
4. **Ensure responsiveness:** Use CSS Grid and media queries to create a layout that adapts seamlessly to various screen sizes, from mobile devices to desktop computers.

Theory

This project utilizes core web development principles:

- **HTML:** The fundamental building block for creating the web page's structure. Tags like <div>, <head>, <body>, <title>, and <meta> are used to define the page's content and metadata.
- **CSS:** Cascading Style Sheets are used to control the visual presentation of the HTML elements. This includes defining colors, fonts, spacing, and the layout. The use of **CSS variables** (--bg-color, --text-color, etc.) is a key technique for implementing a dynamic theme switcher, allowing for easy updates to the entire color scheme.
- **Responsive Design:** This approach ensures the web application looks good and functions well on all devices. The viewport meta tag and CSS Grid are essential for creating a flexible, fluid layout that adjusts to the screen size.

Procedure

1. **HTML Structure:** Begin by writing the basic HTML boilerplate. Define the <head> section with the title and meta viewport tag.
2. **CSS Styling:**
   - Create a <style> block in the HTML or link to an external styles.css file.
   - Define :root CSS variables for both light and dark themes.

    - Create a body selector to apply the default light theme styles.
    - Add a .dark-mode class to the body, which will be applied by JavaScript to override the default variables with dark theme values.
    - Use CSS Grid to create the main layout with a sidebar and content area.
    - Style all elements, including the header, sidebar navigation (<ul>, <li>, <a>), and the welcome card.
3. **JavaScript Functionality:**
    - Add a <script> block to the HTML or link an external file (script.js).
    - Create a function that, when called, toggles the .dark-mode class on the <body> element.
    - Attach this function to a button (e.g., "Switch Theme") using an onclick event listener.
4. **Testing:** Open the index.html file in a web browser. Test the theme-switching button and resize the browser window to confirm the layout is responsive.

Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Admin Dashboard</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" />
  <style>
    /* Base Styles using CSS variables for theme switching */
    :root {
      --bg-color: #f1f5f9;
      --text-color: #333;
      --header-bg: #fff;
      --sidebar-bg: #e2e8f0;
      --card-bg: #fff;
      --shadow-color: rgba(0, 0, 0, 0.1);
    }

    .dark {
      --bg-color: #1e293b;
      --text-color: #f5f5f5;
```

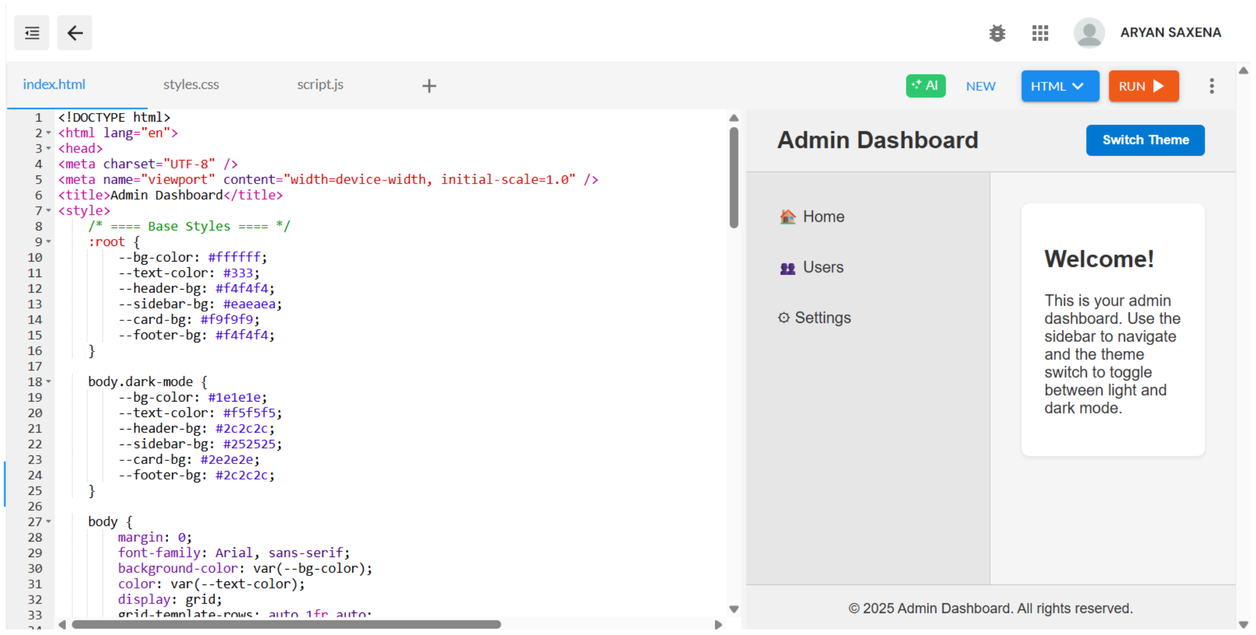
```css
    --header-bg: #334155;
    --sidebar-bg: #475569;
    --card-bg: #334155;
    --shadow-color: rgba(0, 0, 0, 0.5);
}

body {
    background-color: var(--bg-color);
    color: var(--text-color);
    font-family: 'Inter', sans-serif;
    transition: all 0.3s ease;
    margin: 0;
    display: grid;
    grid-template-columns: 250px 1fr;
    height: 100vh;
}

header {
    grid-column: 2 / 3;
    background-color: var(--header-bg);
    box-shadow: 0 2px 4px var(--shadow-color);
}

aside {
    grid-column: 1 / 2;
    background-color: var(--sidebar-bg);
    transition: all 0.3s ease;
    display: flex;
    flex-direction: column;
    justify-content: space-between;
}

.sidebar-nav a {
    display: flex;
    align-items: center;
    padding: 1rem;
    color: var(--text-color);
    text-decoration: none;
    transition: background-color 0.2s;
```

```
        }

        .sidebar-nav a:hover {
            background-color: rgba(0, 0, 0, 0.1);
        }

        .sidebar-nav i {
            width: 30px;
        }

        main {
            grid-column: 2 / 3;
            padding: 2rem;
        }

        .card {
            background-color: var(--card-bg);
            box-shadow: 0 4px 6px var(--shadow-color);
            border-radius: 0.5rem;
            padding: 1.5rem;
            text-align: center;
            transition: all 0.3s ease;
        }

        /* Responsive design */
        @media (max-width: 768px) {
            body {
                grid-template-columns: 1fr;
            }
            aside {
                display: none; /* Hide sidebar by default on small screens */
            }
            header, main {
                grid-column: 1 / 2;
            }
        }
    </style>
</head>
<body class="dark">
```

```
    <header class="flex justify-end items-center p-4">
        <button id="theme-toggle" class="bg-blue-500 hover:bg-blue-600 text-white
font-bold py-2 px-4 rounded-lg shadow-md transition duration-300 ease-in-out">
            Switch Theme
        </button>
    </header>

    <aside>
        <div class="p-4">
            <h1 class="text-2xl font-bold mb-6">Admin Dashboard</h1>
            <nav class="sidebar-nav">
                <ul>
                    <li><a href="#"><i class="fas fa-home"></i> Home</a></li>
                    <li><a href="#"><i class="fas fa-users"></i> Users</a></li>
                    <li><a href="#"><i class="fas fa-cogs"></i> Settings</a></li>
                </ul>
            </nav>
        </div>
        <footer class="text-center p-4 text-sm opacity-50">
            © 2023 Admin Dashboard. All rights reserved.
        </footer>
    </aside>

    <main>
        <div class="card max-w-sm mx-auto">
            <h2 class="text-3xl font-bold mb-4">Welcome!</h2>
            <p>This is your admin dashboard. Use the sidebar to navigate and the theme
switch to toggle between light and dark mode.</p>
        </div>
    </main>

    <script>
        const themeToggleBtn = document.getElementById('theme-toggle');
        const body = document.body;

        themeToggleBtn.addEventListener('click', () => {
            body.classList.toggle('dark');
        });
    </script>
```

```
</body>
</html>
```

Output



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Admin Dashboard</title>
<style>
    /* ==== Base Styles ==== */
    :root {
        --bg-color: #ffffff;
        --text-color: #333;
        --header-bg: #f4f4f4;
        --sidebar-bg: #eaeaea;
        --card-bg: #f9f9f9;
        --footer-bg: #f4f4f4;
    }

    body.dark-mode {
        --bg-color: #1e1e1e;
        --text-color: #f5f5f5;
        --header-bg: #2c2c2c;
        --sidebar-bg: #252525;
        --card-bg: #2e2e2e;
        --footer-bg: #2c2c2c;
    }

    body {
        margin: 0;
        font-family: Arial, sans-serif;
        background-color: var(--bg-color);
        color: var(--text-color);
        display: grid;
        grid-template-rows: auto 1fr auto;
```

Admin Dashboard

Switch Theme

🏠 Home

👥 Users

⚙ Settings

Welcome!

This is your admin dashboard. Use the sidebar to navigate and the theme switch to toggle between light and dark mode.

© 2025 Admin Dashboard. All rights reserved.

Learning Outcomes

Upon successful completion of this lab, the student will be able to:

1. **Develop a web page layout:** Create a multi-column layout using modern CSS techniques like Grid, ensuring proper spacing and alignment of elements.
2. **Implement dynamic styling:** Utilize CSS variables to create a theme switcher, demonstrating an understanding of how to manage and change styles dynamically.
3. **Build a responsive interface:** Design a web page that adapts to different screen sizes, understanding the importance of the viewport meta tag and CSS media queries.