

## EXPERIMENT-3

### AIM

To design and build a professional, responsive admin dashboard interface using HTML, CSS Grid layout, and implement a theme switching feature (light and dark modes).

### OBJECTIVE

This practice focuses on advanced layout skills using CSS Grid and demonstrates the adaptability of a user interface through theme switching without reloading the page.

### THEORY

CSS Grid Layout allows web developers to create complex responsive web design layouts more easily and consistently across browsers. Theming enables switching between light and dark modes, improving accessibility and user experience. JavaScript is used to toggle classes or attributes that change the dashboard's appearance dynamically.

### 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>
  <style>
    :root {
      --bg-color: #ffffff;
      --text-color: #000000;
      --header-bg: #f0f0f0;
      --sidebar-bg: #e0e0e0;
    }

    .dark-mode {
      --bg-color: #1e1e1e;
      --text-color: #ffffff;
      --header-bg: #2e2e2e;
      --sidebar-bg: #3e3e3e;
    }

    body {
```

```
margin: 0;
font-family: Arial, sans-serif;
background-color: var(--bg-color);
color: var(--text-color);
display: grid;
grid-template-areas:
  "header header"
  "sidebar main"
  "footer footer";
grid-template-columns: 200px 1fr;
grid-template-rows: 60px 1fr 40px;
height: 100vh;
}

header {
  grid-area: header;
  background-color: var(--header-bg);
  padding: 10px 20px;
  display: flex;
  justify-content: space-between;
  align-items: center;
}

aside {
  grid-area: sidebar;
  background-color: var(--sidebar-bg);
  padding: 20px;
}

main {
  grid-area: main;
  padding: 20px;
}

footer {
  grid-area: footer;
  background-color: var(--header-bg);
  text-align: center;
  padding: 10px;
}

button.theme-toggle {
  padding: 5px 10px;
  cursor: pointer;
}

a {
  display: block;
  margin: 10px 0;
  color: var(--text-color);
  text-decoration: none;
}
```

```
    a:hover {
      text-decoration: underline;
    }
  </style>
</head>
<body>
  <header>
    <h2>Admin Dashboard</h2>
    <button class="theme-toggle" onclick="toggleTheme()">Toggle Theme</button>
  </header>

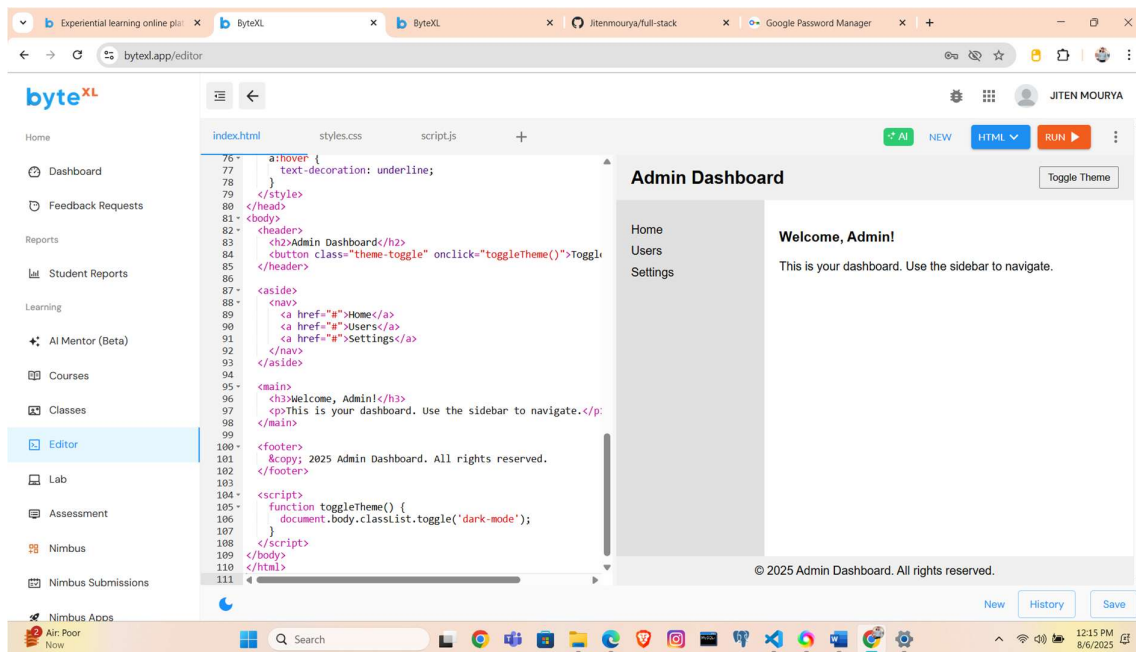
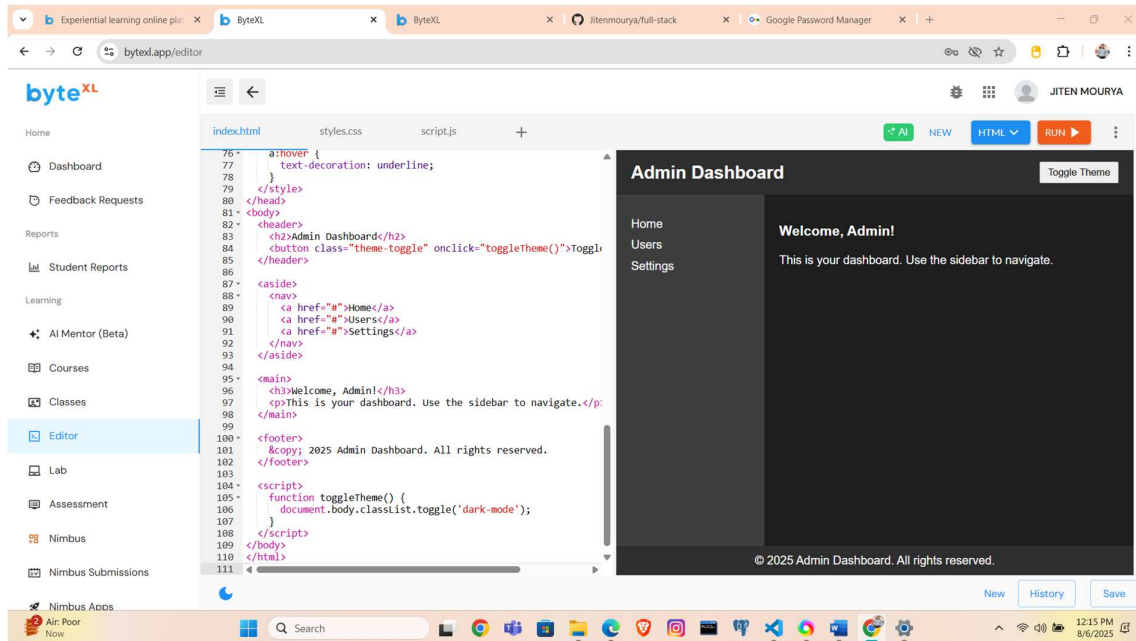
  <aside>
    <nav>
      <a href="#">Home</a>
      <a href="#">Users</a>
      <a href="#">Settings</a>
    </nav>
  </aside>

  <main>
    <h3>Welcome, Admin!</h3>
    <p>This is your dashboard. Use the sidebar to navigate.</p>
  </main>

  <footer>
    &copy; 2025 Admin Dashboard. All rights reserved.
  </footer>

  <script>
    function toggleTheme() {
      document.body.classList.toggle('dark-mode');
    }
  </script>
</body>
</html>
```

# OUTPUT



## LEARNING OUTCOME

After completing this practice, students will be able to:

- Use CSS Grid to design a professional layout.
- Implement responsive web components.
- Add a dynamic theme switching feature using JavaScript.
- Enhance user experience with adaptable design elements.