Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the implementation details for **StyleSwitch - CSS Theme Toggler**, a React-based web application that enables users to switch between multiple CSS themes (light, dark, custom) using Tailwind CSS and CSS variables.

---

1. System Components

Component	Description	Key Responsibilities
ThemeProvider	Context provider for theme state	Manage and provide current theme
ThemeToggle	UI control to switch themes	Trigger theme change
App	Main application container	Compose components, apply theme
ThemeConfig	Theme definitions and CSS variable mapping	Store theme data

---

2. Class/Interface Overview

Name	Type	Relationships	Key Methods/Attributes
ThemeProvider	Function	Wraps App, provides context	`theme` , `setTheme` , `children`
ThemeContext	Context	Used by ThemeToggle, App	`theme` , `setTheme`
ThemeToggle	Function	Consumes ThemeContext	`onClick` , `currentTheme`
ThemeConfig	Object	Used by ThemeProvider	`themes` (light, dark, custom)

Key Methods/Attributes Example:

```
// ThemeProvider
const [theme, setTheme] = useState('light');
<ThemeContext.Provider value={{ theme, setTheme }} />

// ThemeToggle
<button onClick={() => setTheme(nextTheme)}>Toggle Theme</button>
```

---

3. Data Structure Overview

Data Model	Structure / Fields	Purpose
ThemeConfig	`{ [themeName]: { [var]: value } }`	Store CSS variable mappings

ThemeContext	`{ theme: string, setTheme: function }`	Provide theme state globally

Example:

```
const ThemeConfig = {
  light:  { '--bg': '#fff', '--text': '#000' },
  dark:   { '--bg': '#000', '--text': '#fff' },
  custom: { '--bg': '#f0e', '--text': '#333' }
};
```

---

4. Algorithms/Logic

Theme Switching Flow (Pseudocode):

```
On ThemeToggle click:
  Get current theme from context
  Determine next theme (cycle through available themes)
  setTheme(nextTheme)
  For each CSS variable in ThemeConfig[nextTheme]:
    Set variable on document root (document.documentElement.style)
```

---

5. Error Handling

Scenario	Handling Approach
Invalid theme selection	Fallback to default theme
ThemeConfig missing/undefined	Use hardcoded fallback values
Context not available	Show error message or default UI

End of Document