



# Product Requirements & Specification Document

---

## Project Name

**StyleSwitch - CSS Theme Toggler**

## Description

StyleSwitch is a web application that enables users to toggle between multiple CSS themes (light, dark, and custom) using React, JavaScript, and Tailwind CSS. The app demonstrates the use of CSS selectors and variables for dynamic theming. Targeted for startup and educational use cases.

---

## 1. Goals & Objectives

Goal	Description
Theme Switching	Allow users to switch between light, dark, and custom themes
Educational Demonstration	Showcase CSS selectors and variables in a React/Tailwind context
Responsive UI	Ensure usability across devices
Simple Integration	Easy to integrate and extend for other projects

---

## 2. Features

Feature	Description
Theme Toggle UI	Button or switch to select between themes
Theme Persistence	Remember user's theme preference (localStorage)
Custom Theme Support	Allow users to define and apply a custom theme (color pickers or presets)
Live Theme Preview	Instantly update UI on theme change
Accessibility	Ensure sufficient color contrast and keyboard navigation
Educational Tooltips	Optional tooltips explaining CSS selectors/variables usage

---

## 3. User Stories

As a...	I want to...	So that...
User	Switch between light, dark, and custom themes	I can personalize my viewing experience
User	See changes instantly	I can preview themes in real time
Developer	Understand theme logic	I can learn about CSS variables/selectors
User	Have my theme remembered	I don't have to reselect on each visit



## 4. Technical Requirements

### 4.1 Frontend

Requirement	Specification
Framework	React (functional components, hooks)
Styling	Tailwind CSS (with custom configuration)
State Management	React useState/useContext
Theme Persistence	localStorage
Accessibility	WCAG 2.1 AA compliance

### 4.2 Theming

- Use CSS variables for theme colors
- Apply variables via Tailwind's `:root` and custom classes
- Example:

```
:root {
  --color-bg: #fff;
  --color-text: #111;
}

.theme-dark {
  --color-bg: #111;
  --color-text: #fff;
}
```

### 4.3 Theme Toggle Logic

```
// Pseudocode
const [theme, setTheme] = useState(getThemeFromStorage() || 'light');
useEffect(() => {
  document.documentElement.className = `theme-${theme}`;
  saveThemeToStorage(theme);
}, [theme]);
```

## 5. UI/UX Specifications

Element	Specification
Theme Toggle	Prominent switch/button, accessible via keyboard
Theme Preview	Main content area updates instantly on theme change
Custom Theme	Modal or panel for color selection (optional presets)
Tooltips	Optional, togglable educational tooltips



Responsiveness	Mobile-first, adapts to all screen sizes
----------------	--

## 6. Non-Functional Requirements

Requirement	Specification
Performance	Theme switch < 100ms
Security	No sensitive data stored/transmitted
Maintainability	Modular, well-documented code
Extensibility	Easy to add new themes

## 7. Out of Scope

- Backend/API integration
- User authentication
- Advanced theme editor (beyond color pickers/presets)

## 8. Acceptance Criteria

ID	Criteria
AC1	User can toggle between light, dark, and custom themes
AC2	Theme changes are applied instantly and persist on reload
AC3	UI is responsive and accessible
AC4	Code demonstrates use of CSS variables/selectors with Tailwind
AC5	Optional: Custom theme can be defined via color pickers

## 9. Milestones & Timeline

Milestone	Description	Target Date
UI Prototype	Basic layout and theme toggle	Week 1
Theme Logic Integration	CSS variables, Tailwind config	Week 2
Custom Theme Support	Color picker/presets	Week 3
Accessibility & Testing	WCAG compliance, cross-browser test	Week 4
Documentation & Release	Code comments, README, deployment	Week 5

## 10. Dependencies



Dependency	Version/Notes
React	18.x
Tailwind CSS	3.x
JavaScript	ES6+

---

## 11. Risks & Mitigations

Risk	Mitigation
Browser compatibility	Test on major browsers
Accessibility gaps	Use tools (axe, Lighthouse)
Theme conflicts	Namespace CSS variables/classes

---

## 12. Appendix

- [Tailwind CSS Theming Docs](#)
  - [React useContext Docs](#)
  - [WCAG Guidelines](#)
-