

# Theme System Documentation

---

## Overview

---

The DeliveryNext application now features a comprehensive theme switching system that allows users to toggle between two distinct themes:

1. **Original Theme** (Default) - The original orange/red gradient design
2. **Netflix Theme** - A dark, Netflix-inspired design with signature red accents

## Features

---

### Theme Switching

- **Easy Toggle:** Theme toggle button in the header allows instant switching
- **Persistent:** Theme preference is saved to localStorage
- **Seamless:** Smooth transitions between themes

### Netflix Theme Characteristics

#### Colors

- **Primary Red:** #E50914 (Netflix signature red)
- **Dark Red:** #B20710 (Hover states)
- **Black:** #000000 (Main background)
- **Dark Gray:** #141414 (Card backgrounds)
- **Medium Gray:** #2F2F2F (Secondary elements)
- **Light Gray:** #808080 (Muted text)

#### Typography

- Netflix Sans-inspired font stack
- Bold, impactful headings
- Tight letter spacing (-0.02em)
- Heavy font weights (600+)

#### Components

- **NetflixHero:** Full-screen hero banner with gradient overlays
- **NetflixCarousel:** Horizontal scrolling content carousel
- **NetflixCard:** Hover-scaling cards with image overlays
- **NetflixFeatureCard:** Feature cards with red accent icons

#### Design Language

- Dark backgrounds with high contrast
- Red accent color throughout
- Smooth hover animations and transitions
- Card scaling effects on hover
- Horizontal content carousels
- Dramatic shadows and overlays

# Implementation

## File Structure

```
src/
├── contexts/
│   └── ThemeContext.tsx      # Theme state management
├── components/
│   ├── theme/
│   │   └── ThemeToggle.tsx  # Theme toggle button
│   └── netflix/
│       ├── NetflixHero.tsx  # Netflix-style hero section
│       ├── NetflixCarousel.tsx # Horizontal scrolling carousel
│       ├── NetflixCard.tsx   # Netflix-style content card
│       └── NetflixFeatureCard.tsx # Netflix-style feature card
└── app/
    ├── globals.css          # Theme CSS variables
    ├── layout.tsx           # ThemeProvider wrapper
    └── page.tsx             # Theme-aware home page
```

## Usage

### Using the Theme Context

```
import { useTheme } from '@contexts/ThemeContext';

function MyComponent() {
  const { theme, toggleTheme, setTheme } = useTheme();

  return (
    <div className={theme === 'netflix' ? 'bg-black' : 'bg-white'}>
      <button onClick={toggleTheme}>Toggle Theme</button>
    </div>
  );
}
```

### Theme-Aware Styling

```
const isNetflix = theme === 'netflix';

<div className={` ${isNetflix ? 'bg-black text-white' : 'bg-white text-black'} `}>
  Content
</div>
```

## Using Netflix Components

```
import { NetflixCarousel } from '@components/netflix/NetflixCarousel';
import { NetflixCard } from '@components/netflix/NetflixCard';

const items = [
  { id: '1', title: 'Item 1', description: 'Description' }
];

<NetflixCarousel
  title="Featured Items"
  items={items}
  renderItem={(item) => (
    <NetflixCard
      title={item.title}
      description={item.description}
      onClick={() => {}}
    />
  )}
/>
```

## CSS Variables

Both themes use CSS variables for consistent theming:

```
/* Original Theme */
--primary: oklch(0.7686 0.1647 70.0804);
--background: oklch(1.0000 0 0);

/* Netflix Theme */
--primary: #E50914;
--background: #000000;
```

## Tailwind Configuration

Netflix colors are available in Tailwind:

```
<div className="bg-netflix-red text-netflix-white">
  Netflix styled content
</div>
```

## Best Practices

1. **Always use the theme context** for conditional styling
2. **Leverage CSS variables** for consistent theming
3. **Use Netflix components** when in Netflix theme for authentic experience
4. **Test both themes** when adding new features
5. **Maintain accessibility** in both themes (contrast ratios, etc.)

## Future Enhancements

- Additional theme options (e.g., Dark Mode, Light Mode variants)
- Per-page theme preferences
- Theme-specific animations

- More Netflix-inspired components (modals, forms, etc.)
- Theme preview before switching
- Custom theme builder

## Technical Details

---

### Theme Persistence

- Stored in `localStorage` as `delivery-theme`
- Loaded on initial mount
- Applied via document class ( `theme-original` , `theme-netflix` )

### Performance

- Theme context uses React Context API
- Minimal re-renders with proper memoization
- CSS variables for instant theme switching
- No flash of unstyled content (FOUC)

### Browser Support

- Modern browsers (Chrome, Firefox, Safari, Edge)
- CSS variables support required
- `localStorage` support required

## Troubleshooting

---

### Theme not persisting

- Check `localStorage` is enabled
- Verify `delivery-theme` key exists
- Clear cache and reload

### Styles not applying

- Ensure `ThemeProvider` wraps your app
- Check CSS variable definitions
- Verify Tailwind config includes Netflix colors

### Components not rendering

- Import from correct paths
- Check theme context is available
- Verify component props are correct