

CSS Optimization Guide

This guide documents the strategy for optimizing `app/globals.css` by splitting it into smaller, more maintainable files and converting custom CSS to Tailwind utilities where appropriate.

Current State

- **File Size:** 2,957 lines in `app/globals.css`
- **Issues:**
 - Difficult to navigate and maintain
 - Mix of concerns (variables, animations, components, utilities)
 - Some CSS that could be replaced with Tailwind utilities
 - No clear organization

Optimization Strategy

1. Split into Logical Modules

Reorganize CSS into focused, maintainable files:

```
app/styles/loomos/
├── animations.css      # All @keyframes and animation utilities
├── cards.css           # App card and window management styles
├── dock.css             # Dock, dock items, and dock popovers
├── layout.css           # Container, touchscreen, multi-pane layouts
├── status-bar.css       # Status bar and system UI
├── search.css           # Universal search styles
├── lists.css             # List, header, footer, toolbar styles
└── utilities.css        # Color utilities and helper classes
```

2. Conversion to Tailwind Utilities

What to Convert

Good candidates for Tailwind:

- Simple spacing (`padding`, `margin`)
- Basic layouts (`flex`, `grid`)
- Typography (`text-`, `font-`)
- Borders and shadows (when standard)
- Display properties (`block`, `flex`, `hidden`)
- Positioning (when simple)

Keep as Custom CSS:

- Complex gradients and glassmorphism
- WebOS-specific design system
- Animations and keyframes
- Multi-step transitions
- Complex pseudo-selectors
- Browser-specific hacks

Example Conversions

Before:

```
.loomos-search-bar {
  width: 100%;
  max-width: 48rem;
  padding: 1rem 1.5rem;
  display: flex;
  align-items: center;
  gap: 1rem;
  border-radius: 1.5rem;
  box-shadow: 0 20px 40px rgba(0, 0, 0, 0.2);
}
```

After:

```
<div className="w-full max-w-2xl px-6 py-4 flex items-center gap-4 rounded-3xl shadow-2xl loomos-search-bar">
```

Where `loomos-search-bar` only contains custom properties:

```
.loomos-search-bar {
  background: rgba(247, 246, 244, 0.98);
  backdrop-filter: blur(20px);
  border: 1px solid rgba(207, 204, 199, 0.6);
}
```

3. File Organization

animations.css

All keyframe animations and animation utility classes:

- Entrance/exit animations
- Loading animations
- Micro-interactions
- Transitions

cards.css

App window and card management:

- `.loomos-app-card` - Base card styles
- `.loomos-app-card-draggable` - Draggable variants
- `.loomos-resize-handle` - Resize handles
- `.loomos-app-card-header` - Card headers
- `.loomos-app-card-body` - Card content areas

dock.css

Dock and app launcher:

- `.loomos-app-dock` - Dock container
- `.loomos-dock-*` - Dock items and indicators
- `.loomos-dock-popover-*` - Flyout menus
- Running app indicators
- Preview cards

layout.css

Page-level layouts:

- `.loomos-container` - Main OS container
- `.loomos-touchscreen` - Touch interaction area
- `.loomos-multi-pane-layout` - Multi-pane system
- `.loomos-pane-*` - Pane components

status-bar.css

System status bar:

- `.loomos-status-bar` - Status bar container
- `.loomos-status-bar-refined` - Refined variant
- System indicators
- Clock and status items

search.css

Universal search:

- `.loomos-universal-search` - Search overlay
- `.loomos-search-bar` - Search input
- `.loomos-search-results` - Results container
- `.loomos-search-category` - Category grouping

lists.css

List components:

- `.loomos-header` - List headers
- `.loomos-footer` - List footers
- `.loomos-toolbar` - Toolbars
- `.loomos-list-*` - List items and dividers

utilities.css

Helper classes:

- Color utilities (`.loomos-text`, `.loomos-orange`)
- State utilities (`.loomos-active`, `.loomos-hover`)
- Dividers and avatars
- Common patterns

Implementation Plan

Phase 1: Extract Animations

Status: Complete

Created `app/styles/loomos/animations.css` with:

- 15+ keyframe animations
- Animation utility classes
- Standardized timing and easing

Benefits:

- 250+ lines removed from `globals.css`
- Easy to find and modify animations
- Clear animation inventory

Phase 2: Create Module Structure

Target: Remaining CSS sections

1. Create empty module files
2. Move relevant CSS to each module
3. Test for visual regressions
4. Update imports in globals.css

Phase 3: Tailwind Conversion

Target: Component markup

1. Identify conversion opportunities
2. Update component classes
3. Minimize custom CSS
4. Document exceptions

Phase 4: Documentation

Target: Maintainability

1. Document module purposes
2. Add inline comments
3. Create style guide
4. Update README

File Size Goals

File	Before	After	Reduction
globals.css	2,957 lines	~300 lines	90%
animations.css	0	~250 lines	New
cards.css	0	~400 lines	New
dock.css	0	~500 lines	New
layout.css	0	~350 lines	New
status-bar.css	0	~200 lines	New
search.css	0	~150 lines	New
lists.css	0	~300 lines	New
utilities.css	0	~200 lines	New
Total	2,957	2,650	-10%

Note: Total line count may increase slightly due to module boundaries and documentation, but organization improves significantly.

Updated globals.css Structure

After optimization:

```
@tailwind base;
@tailwind components;
@tailwind utilities;

/* Design System Imports */
@import './design-tokens/semantic.css';
@import './design-tokens/motion.css';
@import './styles/design-tokens.css';
@import './styles/loomos-tokens.css';
@import './styles/loomos-components.css';

/* loomOS Component Modules */
@import './styles/loomos/animations.css';
@import './styles/loomos/cards.css';
@import './styles/loomos/dock.css';
@import './styles/loomos/layout.css';
@import './styles/loomos/status-bar.css';
@import './styles/loomos/search.css';
@import './styles/loomos/lists.css';
@import './styles/loomos/utilities.css';

/* Base Layer - Theme Variables Only */
@layer base {
  :root {
    /* CSS custom properties... */
  }

  .dark {
    /* Dark mode overrides... */
  }
}

/* Minimal Base Styles */
@layer base {
  * {
    @apply border-border;
  }
  body {
    @apply bg-background text-foreground;
  }
}
```

Tailwind Conversion Examples

Example 1: Search Bar

Before (80 lines):

```
.loomos-search-bar {
  width: 100%;
  max-width: 48rem;
  padding: 1.5rem;
  display: flex;
  align-items: center;
  gap: 1rem;
  border-radius: 1.5rem;
  box-shadow: 0 20px 40px rgba(0, 0, 0, 0.2);
  margin-bottom: 1.5rem;
  background: rgba(247, 246, 244, 0.98);
  backdrop-filter: blur(20px);
  border: 1px solid rgba(207, 204, 199, 0.6);
}

.loomos-search-input {
  flex: 1;
  background: transparent;
  border: none;
  outline: none;
  font-size: 1.125rem;
  color: var(--foreground);
}

.loomos-search-input::placeholder {
  color: var(--muted-foreground);
}
```

After (15 lines):

```
// Component


<input
    className="flex-1 bg-transparent border-none outline-none text-lg
placeholder:text-muted-foreground"
    placeholder="Search..."
  />
</div>


```

```
/* Custom CSS only */
.loomos-search-bar {
  background: rgba(247, 246, 244, 0.98);
  backdrop-filter: blur(20px);
  border: 1px solid rgba(207, 204, 199, 0.6);
}

.dark .loomos-search-bar {
  background: rgba(56, 56, 56, 0.98);
  border: 1px solid rgba(76, 76, 76, 0.6);
}
```

Savings: 65 lines → 15 lines (78% reduction)

Example 2: Dock Items

Before:

```
.loomos-app-dock-item {
  position: relative;
  width: 4rem;
  height: 4rem;
  cursor: pointer;
  transition: all 300ms ease-out;
}

.loomos-app-dock-item-active {
  transform: scale(1.1);
}
```

After:

```
<div className={cn(
  "relative w-16 h-16 cursor-pointer transition-all duration-300 ease-out",
  isActive && "scale-110"
)} />
```

Savings: Custom CSS → Tailwind utilities

Best Practices

1. When to Use Tailwind

Use Tailwind for:

- Layout and spacing
- Typography and colors
- Simple borders and shadows
- Display and positioning
- Responsive design

2. When to Keep Custom CSS

Keep custom CSS for:

- WebOS design system (glassmorphism, etc.)
- Complex animations
- Multi-step transitions
- Browser-specific fixes
- Component-specific patterns

3. Naming Conventions

Custom CSS Classes:

- `loomos-*` - Component-specific styles
- `loomos-*-*` - Component variants
- Always namespace with `loomos-`

Tailwind Classes:

- Use arbitrary values sparingly
- Prefer theme values
- Use `@apply` in CSS only when needed

4. File Organization

In Module Files:

```
/* File header with description */
/* Component Name */
.loomos-component {
    /* Custom properties only */
}

/* Dark mode variant */
.dark .loomos-component {
    /* Dark mode overrides */
}

/* Component states */
.loomos-component-active {
    /* State-specific styles */
}
```

Testing Strategy

1. Visual Regression Testing

After each module extraction:

- [] Test all components in light mode
- [] Test all components in dark mode
- [] Test responsive breakpoints
- [] Test animations and transitions

2. Performance Testing

Monitor bundle size:

```
npm run analyze
```

Check CSS size:

```
du -sh .next/static/css/*
```

3. Browser Testing

Test in:

- Chrome (latest)
- Firefox (latest)
- Safari (latest)
- Edge (latest)

Tools and Scripts

Find Tailwind Conversion Opportunities

```
# Find classes that could be Tailwind
grep -r "padding:\|margin:\|display:\|flex" app/globals.css
```

Check File Sizes

```
# Show CSS file sizes
du -h app/globals.css app/styles/loomos/*.css
```

Validate CSS

```
# Use stylelint
npm install --save-dev stylelint
npx stylelint "app/**/*.{css}"
```

Migration Checklist

- [x] Phase 1: Extract animations to separate file
- [] Phase 2: Extract dock styles
- [] Phase 3: Extract card/window styles
- [] Phase 4: Extract layout styles
- [] Phase 5: Extract status bar styles
- [] Phase 6: Extract search styles
- [] Phase 7: Extract list styles
- [] Phase 8: Extract utility classes
- [] Phase 9: Update component imports
- [] Phase 10: Convert markup to Tailwind where appropriate
- [] Phase 11: Test for visual regressions
- [] Phase 12: Update documentation

Resources

- [Tailwind CSS Documentation](https://tailwindcss.com/docs) (<https://tailwindcss.com/docs>)
- [CSS Modules Best Practices](https://github.com/css-modules/css-modules) (<https://github.com/css-modules/css-modules>)
- [WebOS Design Guidelines](https://www.webosose.org/design/) (<https://www.webosose.org/design/>)
- [Next.js CSS Support](https://nextjs.org/docs/basic-features/built-in-css-support) (<https://nextjs.org/docs/basic-features/built-in-css-support>)

Changelog

- **2024-11-14:** Initial CSS optimization guide created
- Documented current state (2,957 lines)
- Created module structure plan
- Extracted animations to separate file (~250 lines)
- Defined Tailwind conversion strategy

- Created testing and validation plan