loomOS Styling Playground Guide

Welcome to your loomOS styling playground! This is a full copy of the loomOS project where you can freely experiment with styling changes before committing them to a PR.

📁 Project Structure Overview

Main Dashboard Location

The main dashboard you'll be styling is located at:
- **Main Dashboard Page**: `app/dashboard/page.tsx`
- **Dashboard Layout**: `app/dashboard/layout.tsx`

Key Component Directories

1. Dashboard Components (`components/dashboard/`)

- `sidebar.tsx` - Main navigation sidebar
- `widgets/` - Dashboard widget components
- `chart.tsx` - Chart components for data visualization

2. WebOS Components (`components/webos/`)

Modern OS-like interface components:
- `desktop-top-bar.tsx` - Top navigation bar
- `app-dock.tsx` - Application dock/launcher
- `desktop-widget-manager.tsx` - Widget system
- `loomos-*.tsx` - Various layout and UI components
- Card system components (carousel, stack views)

3. UI Primitives (`components/ui/`)

shadcn/ui based components:
- `button.tsx`, `card.tsx`, `dialog.tsx`, etc.
- Foundational UI building blocks

4. Dashboard Pages (`app/dashboard/`)

```
dashboard/
├── page.tsx              # Main dashboard with card carousel
├── layout.tsx            # Dashboard layout with top bar & search
├── admin/               # Admin pages
├── apps/                # Built-in apps (calendar, notes, etc.)
├── my-community/        # Community features
├── my-household/        # Household management
├── messages/            # Messaging system
├── documents/           # Document management
└── payments/            # Payment features
```

🎨 Styling System

Design Tokens

The project uses a comprehensive design system with CSS variables defined in:
- **Core Tokens**: `design-tokens/core.css`
- **Semantic Tokens**: `design-tokens/semantic.css`
- **Motion Tokens**: `design-tokens/motion.css`

Main Styling Files

1. **Global Styles**: `app/globals.css`
   - Base styles and imports
   - Design token imports
   - Custom animations

2. **Tailwind Configuration**: `tailwind.config.ts`
   - Extended Tailwind with design tokens
   - Custom colors, spacing, shadows
   - Responsive breakpoints

3. **Component Animations**: `app/styles/loomos/animations.css`
   - Card animations
   - Transitions
   - Effects

Current Color Scheme

The dashboard uses a modern, neutral palette with CSS variables:

Primary Colors:
- Background gradient: `linear-gradient(135deg, var(--semantic-bg-muted) 0%, var(--semantic-bg-subtle) 50%, var(--semantic-bg-muted) 100%)`
- Card backgrounds: Various soft colors (#f0f8e8, #e8f8f0, #f8f0e8, etc.)
- Accent colors: Blues (#7a9eb5), Browns (#b58a7a), Greens (#8ab57a)

Text Colors:
- Primary: `var(--semantic-text-primary)`
- Secondary: `var(--semantic-text-secondary)`
- Tertiary: `var(--semantic-text-tertiary)`

Chrome/UI Colors:
- Dark chrome: `var(--chrome-darker)`, `var(--chrome-dark)`
- Text on chrome: `var(--chrome-text)`

🚀 How to Run the Playground

Quick Start

1. **Navigate to the playground directory:**
   `bash`
   ```
      cd /home/ubuntu/code_artifacts/loomOS-styling-playground
   ```

2. **Start the development server:**
   `bash`
   `npm run dev`

3. **Open in browser:**
   `http://localhost:3000`

The dashboard is at: `http://localhost:3000/dashboard`

Note: You may see authentication screens first. For styling work, you can:
- Create a test account
- Or bypass auth by directly modifying the layout temporarily

Alternative: Run on Different Port

If port 3000 is busy:

```
PORT=3001 npm run dev
```

Stop the Server

Press `Ctrl+C` in the terminal where the server is running.

✨ Making Styling Changes

1. Color Scheme Changes

Option A: Modify CSS Variables (Recommended)

Edit `design-tokens/core.css` or `design-tokens/semantic.css`:

```
/* Change the primary background color */
:root {
  --semantic-bg-muted: #f5f5f5; /* New color */
}

/* Change accent colors */
:root {
  --accent-blue: #4a90e2;
  --accent-blue-light: #6fa8ef;
}
```

Option B: Modify Component Styles Directly

Edit `app/dashboard/page.tsx`:

```
// Change card background colors
const cardData = [
  {
    id: 'work-orders',
    title: 'Work Orders',
    type: 'stack',
    color: '#e8f4f8',  // Change this color
    // ...
  }
];

// Change main background
<div
  className="h-full relative overflow-hidden"
  style={{
    background: 'linear-gradient(135deg, #f0f0f0 0%, #e0e0e0 50%, #f0f0f0 100%)',
    // Change this gradient
  }}
>
```

Option C: Use Tailwind Classes

Many components use Tailwind utility classes:

```
// Before
<div className="bg-blue-500 text-white">

// After - using design tokens
<div className="bg-accent-blue text-chrome-text">
```

2. Layout Adjustments

Card Carousel Spacing

Edit `app/dashboard/page.tsx`:

```
// Adjust card spacing in the carousel
style={{
  transform: `translateX(${(index - currentCard) * 450}px)`, // Change 400 to 450
  scale: isCentered ? 1 : 0.85
}}
```

Card Sizes

```
// Change card dimensions
style={{
  width: '450px',   // Change from 400px
  height: '350px',  // Change from 300px
  // ...
}}
```

Top Bar Height

Edit `app/dashboard/layout.tsx`:

```
// Change status bar height
className="fixed top-0 left-0 right-0 h-14"  // Change h-12 to h-14

// Adjust main content padding accordingly
<div className="pt-14 h-screen overflow-hidden">
```

3. Component Redesign

Dock/Launcher

Edit `app/dashboard/page.tsx`:

```
{/* Customize dock appearance */}
<div
  className="fixed bottom-4 left-1/2 transform -translate-x-1/2 px-6 py-3 rounded-2xl"
  style={{
    backgroundColor: 'rgba(255, 255, 255, 0.8)',  // Change transparency
    backdropFilter: 'blur(20px)',                 // Change blur amount
    // ...
  }}
>
```

App Launcher Grid

Edit `app/dashboard/layout.tsx`:

```
{/* Change grid columns */}
<div className="grid grid-cols-8 gap-6">  // Change from grid-cols-6 to grid-cols-8
```

Status Bar

```
{/* Customize status bar */}
<nav
  className="fixed top-0 left-0 right-0 h-12"
  style={{
    backgroundColor: 'var(--chrome-darker)',  // Try different colors
    borderBottom: '1px solid var(--glass-border-light)',
  }}
>
```

4. Typography Changes

Global Font Settings

Edit `tailwind.config.ts`:

```
fontFamily: {
  sans: ['Inter', 'Helvetica Neue', 'Arial', 'sans-serif'],  // Change font
  // or use variables
  sans: ['var(--font-sans)'],
},
```

Component-Specific Fonts

```
// In any component
<h1 style={{ fontFamily: '"SF Pro Display", -apple-system, sans-serif' }}>
  Dashboard
</h1>

// Or with Tailwind
<h1 className="font-display text-4xl">
  Dashboard
</h1>
```

Font Weights

```
// Light weight (common in loomOS)
<span className="font-light">Text</span>

// Change to other weights
<span className="font-normal">Text</span>
<span className="font-medium">Text</span>
<span className="font-semibold">Text</span>
```

5. Responsive Design Improvements

Breakpoints

The project uses standard Tailwind breakpoints:
- `sm` : 640px (mobile landscape)
- `md` : 768px (tablet)
- `lg` : 1024px (desktop)
- `xl` : 1280px (large desktop)
- `2xl` : 1536px (extra large)

Making Components Responsive

```
{/* Mobile-first approach */}
<div className="
  grid
  grid-cols-1        // Mobile: 1 column
  sm:grid-cols-2     // Small: 2 columns
  md:grid-cols-3     // Tablet: 3 columns
  lg:grid-cols-4     // Desktop: 4 columns
  gap-4              // Spacing
">

{/* Responsive text sizes */}
<h1 className="
  text-2xl           // Mobile
  sm:text-3xl        // Small screens
  md:text-4xl        // Tablet
  lg:text-5xl        // Desktop
">

{/* Responsive padding */}
<div className="
  px-4               // Mobile padding
  md:px-8            // Tablet padding
  lg:px-12           // Desktop padding
">
```

Hide/Show on Different Screens

```
{/* Hide on mobile, show on desktop */}
<div className="hidden lg:block">
  Desktop only content
</div>

{/* Show on mobile, hide on desktop */}
<div className="block lg:hidden">
  Mobile only content
</div>
```

6. Animation Adjustments

Card Animations

Edit `app/dashboard/page.tsx`:

```
// Change animation duration
className="absolute transition-all duration-500 ease-out"  // Change to duration-700

// Change animation easing
style={{
  transition: 'all 0.5s cubic-bezier(0.34, 1.56, 0.64, 1)'  // Custom easing
}}
```

Custom Animations

Edit `app/styles/loomos/animations.css` or add to component:

```
@keyframes slideIn {
  from {
    transform: translateX(-100%);
    opacity: 0;
  }
  to {
    transform: translateX(0);
    opacity: 1;
  }
}

.animate-slide-in {
  animation: slideIn 0.5s ease-out;
}
```

🎯 Common Styling Tasks

Task 1: Change Dashboard Background

File: `app/dashboard/page.tsx` or `app/dashboard/layout.tsx`

```
// Find the main container div and modify:
<div
  className="h-full relative overflow-hidden"
  style={{
    background: 'linear-gradient(to bottom, #f8f9fa, #e9ecef)',  // New gradient
    // OR solid color
    backgroundColor: '#f5f5f5',
  }}
>
```

Task 2: Customize Card Colors

File: `app/dashboard/page.tsx`

```
// Modify the cardData array
const cardData = [
  {
    id: 'work-orders',
    title: 'Work Orders',
    type: 'stack',
    color: '#e3f2fd',  // Change to Material Blue Light
    // ...
  },
  {
    id: 'browser',
    title: 'Browser',
    type: 'single',
    color: '#f3e5f5',  // Change to Material Purple Light
    // ...
  },
  // ... update all cards
];
```

Task 3: Adjust Border Radius (Roundness)

File: `tailwind.config.ts`

```
borderRadius: {
  'DEFAULT': '12px',  // Change from 6px to 12px for more rounded
  'lg': '16px',       // Change from 12px
  'xl': '24px',       // Change from 16px
  // ...
}
```

Or directly on components:

```
<div className="rounded-3xl">  // Change from rounded-2xl
```

Task 4: Change Top Bar Color

File: `app/dashboard/layout.tsx`

```
<nav
  style={{
    backgroundColor: '#1e1e1e',  // Change to darker
    // OR use CSS variable
    backgroundColor: 'var(--chrome-darker)',
  }}
>
```

Task 5: Modify Shadows

File: `tailwind.config.ts` or directly on components

```
// In config
boxShadow: {
  'card': '0 4px 16px rgba(0, 0, 0, 0.1)',  // Lighter shadow
}
```

```
// On component
<div style={{ boxShadow: '0 10px 40px rgba(0, 0, 0, 0.2)' }}>
```

Task 6: Update Icon Colors

File: Any component with icons (e.g., `app/dashboard/page.tsx`)

```
// Before
<Mail className="w-6 h-6" style={{ color: 'var(--semantic-text-primary)' }} />

// After
<Mail className="w-6 h-6 text-blue-500" />
// OR
<Mail className="w-6 h-6" style={{ color: '#4a90e2' }} />
```

🔍 Finding What to Modify

1. Use Browser DevTools

- Right-click on any element → "Inspect"
- See the computed styles and CSS classes

- Modify values in DevTools to experiment
- Copy successful changes to your code

2. Search for Specific Elements

```bash
# Find where a specific color is used
grep -r "#7a9eb5" app/ components/

# Find all files using a specific component
grep -r "StatusBar" app/ components/

# Find CSS variable usage
grep -r "var(--chrome-darker)" app/ components/
```

3. Component Hierarchy

Use React DevTools browser extension to:
- See component structure
- Find component file locations
- Inspect props and state

📝 Best Practices

1. Use CSS Variables First

Instead of hardcoding colors, use the design system:

```jsx
// ❌ Not recommended
<div style={{ backgroundColor: '#f0f0f0' }}>

// ✅ Recommended
<div style={{ backgroundColor: 'var(--bg-surface)' }}>
```

2. Keep Changes Organized

- Group related changes together
- Comment your modifications
- Use git to track changes:
  `bash`
  `  git diff app/dashboard/page.tsx`

3. Test Responsiveness

Always check your changes on different screen sizes:
- Mobile: 375px width
- Tablet: 768px width
- Desktop: 1440px width

Use browser DevTools responsive mode (Ctrl+Shift+M / Cmd+Shift+M)

4. Document Your Changes

Keep notes on what you modified:

```
// STYLING CHANGE: Updated card background colors for better contrast
color: '#e8f4f8',  // Changed from #f0f8e8
```

5. Commit Frequently

```
git add .
git commit -m "styling: updated dashboard card colors"
```

🐛 Troubleshooting

Server Won't Start

```
# Clear cache and reinstall
rm -rf node_modules .next
npm install
npm run dev
```

Changes Not Showing

1. **Hard refresh**: Ctrl+Shift+R (Windows/Linux) or Cmd+Shift+R (Mac)
2. **Clear Next.js cache**: Delete `.next` folder
3. **Check file path**: Ensure you're editing the correct file
4. **Check syntax**: Look for console errors

CSS Not Applying

1. **Check specificity**: More specific selectors override general ones
2. **Check Tailwind purge**: Ensure your classes are in the content config
3. **Use `!important`** (last resort):
   `tsx`
   `  style={{ color: '#000 !important' }}`

TypeScript Errors

If you see TS errors but want to test styling:

```
// @ts-ignore
<Component newProp="value" />
```

Or temporarily disable type checking:

```
# In next.config.js
typescript: {
  ignoreBuildErrors: true,
}
```

📚 Additional Resources

Design Inspiration

- **webOS Design**: Look at Palm webOS, LG webOS interfaces

- **macOS Big Sur**: Glassmorphism, card-based UI
- **iPadOS**: Multitasking, widget layouts
- **Material Design 3**: Color systems, motion

Useful Tools

- **Color Palette Generators**:
- Coolors.co (https://coolors.co/)
- Adobe Color (https://color.adobe.com/)
- Material Palette (https://www.materialpalette.com/)
- **Gradient Generators**:
- CSS Gradient (https://cssgradient.io/)
- Gradient Hunt (https://gradienthunt.com/)
- **Shadow Generators**:
- Box Shadow CSS Generator (https://html-css-js.com/css/generator/box-shadow/)
- Smooth Shadow (https://shadows.brumm.af/)

Documentation

- **Next.js**: nextjs.org/docs (https://nextjs.org/docs)
- **Tailwind CSS**: tailwindcss.com/docs (https://tailwindcss.com/docs)
- **shadcn/ui**: ui.shadcn.com (https://ui.shadcn.com)
- **Framer Motion**: framer.com/motion (https://www.framer.com/motion/)

🎨 Example Styling Scenarios

Scenario 1: Dark Mode Dashboard

```
// app/dashboard/page.tsx
<div
  style={{
    background: 'linear-gradient(135deg, #1a1a1a 0%, #2d2d2d 50%, #1a1a1a 100%)',
  }}
>
```

```
// Update card colors to dark variants
const cardData = [
  {
    color: '#2d2d2d',  // Dark gray
    // ...
  }
];
```

Scenario 2: Vibrant Color Scheme

```
// Use bold, saturated colors
const cardData = [
  { id: 'work-orders', color: '#FF6B6B', /* ... */ },  // Coral
  { id: 'browser', color: '#4ECDC4', /* ... */ },      // Teal
  { id: 'mail', color: '#45B7D1', /* ... */ },         // Sky Blue
  { id: 'calendar', color: '#FFA07A', /* ... */ },     // Salmon
];
```

Scenario 3: Minimalist Design

```
// Reduce shadows
style={{
  boxShadow: '0 1px 3px rgba(0,0,0,0.08)',  // Subtle shadow
  border: '1px solid #e0e0e0',              // Add border
}}

// Use more whitespace
className="px-12 py-8"  // Increase from px-8 py-6

// Simplify colors
color: '#ffffff',  // Pure white backgrounds
```

Scenario 4: Glassmorphism Effect

```
<div
  style={{
    backgroundColor: 'rgba(255, 255, 255, 0.1)',
    backdropFilter: 'blur(20px)',
    border: '1px solid rgba(255, 255, 255, 0.2)',
    boxShadow: '0 8px 32px rgba(0, 0, 0, 0.1)',
  }}
>
```

🚀 Next Steps

1. **Explore the codebase**: Open files and see how components are structured
2. **Make small changes**: Start with simple color modifications
3. **Test frequently**: Run the dev server and see your changes live
4. **Document your work**: Keep track of what you like and don’t like
5. **Create a branch**: When ready to commit, create a feature branch
6. **Prepare PR**: Document your changes for the team

📞 Getting Help

If you encounter issues:
1. Check the console for error messages
2. Review this guide’s Troubleshooting section
3. Check the main project README at the root
4. Review existing documentation in the `docs/` folder

Happy Styling! 🎨

Remember: This is your playground - experiment freely! All changes are isolated here and won’t affect the main repository until you create a PR.