🎨 loomOS Styling Playground - Live Access Guide

✅ Your Playground is Now Live!

Dashboard URL: `http://localhost:3000/dashboard`

The server is running and the dashboard is accessible without authentication. You can now start experimenting with styling changes!

🚀 Quick Start - Making Your First Style Change

Example 1: Change Card Background Color

1. **Open the file:** `app/dashboard/page.tsx`
2. **Find the cardData array** (around line 13-30)
3. **Modify a card's color:**

```
const cardData = [
  {
    id: 'work-orders',
    title: 'Pending',
    count: 5,
    color: '#FFE5E5', // ← Change this to any color you want!
    icon: 'clipboard-list',
    type: 'single'
  },
  // ... other cards
];
```

1. **Save the file** - The browser will automatically refresh and show your changes!

Example 2: Change Dashboard Background

1. **Open:** `app/dashboard/page.tsx`
2. **Find the main container** (around line 600+)
3. **Modify the background gradient:**

```
style={{
  background: 'linear-gradient(to bottom, #f0f9ff, #e0f2fe)', // ← Your custom gradient
}}
```

Example 3: Change Top Bar Style

1. **Open:** `app/dashboard/layout.tsx`
2. **Find SystemStatusBar styling** (around line 807)
3. **The top bar is rendered by the SystemStatusBar component**

To customize it further, look at the top bar background in the layout:

```
style={{
  background: 'var(--bg-primary)',  // ← Customize this
  borderBottom: '1px solid var(--border-primary)'
}}
```

Example 4: Adjust Card Size

1. **Open:** `app/dashboard/page.tsx`
2. **Find the card container style** (around line 430+)
3. **Modify dimensions:**

```
style={{
  width: '450px',   // ← Change from default
  height: '350px',  // ← Change from default
}}
```

📂 Key Files Reference

Core Styling Files

File	Purpose	Common Changes
`app/dashboard/page.tsx`	Main dashboard with cards, carousel	Card colors, sizes, layouts, data
`app/dashboard/layout.tsx`	Top bar, app launcher, global search	Top bar styling, layout structure
`app/globals.css`	Global styles, CSS variables	Colors, fonts, spacing tokens
`tailwind.config.ts`	Tailwind customization	Custom colors, breakpoints
`design-tokens/core.css`	Base design tokens	Core color palette, spacing scale
`design-tokens/semantic.css`	Semantic tokens	Component-specific colors

🎯 Real-Time Development Workflow

1. Open Your Editor

Edit files directly in our code editor interface (the files will be shown in the UI).

2. Make Changes

Modify any styling in the files listed above.

3. Save

Save your changes (the file is automatically saved in our editor).

4. See Results Instantly

- Next.js Hot Module Replacement (HMR) will automatically refresh the browser
- Changes appear in seconds without manual refresh
- If changes don't appear, try a hard refresh: `Ctrl+Shift+R` (or `Cmd+Shift+R` on Mac)

🎨 Common Styling Tasks

Change All Card Colors at Once

File: `app/dashboard/page.tsx`

```
const cardData = [
  { id: 'work-orders', color: '#your-color-1', ... },
  { id: 'browser', color: '#your-color-2', ... },
  { id: 'mail', color: '#your-color-3', ... },
  { id: 'calendar', color: '#your-color-4', ... },
];
```

Modify the Search Bar

File: `app/dashboard/layout.tsx`

Look for the `GlobalSearch` component and customize its styling.

Customize the Dock

File: `app/dashboard/page.tsx`

Search for the dock styling (around line 580+) and modify:
- Background color
- Icon sizes
- Positioning
- Spacing

Change Font Styles

File: `app/globals.css`

Modify font-related CSS variables:

```css
:root {
  --font-sans: 'Your Font', sans-serif;
  --font-display: 'Your Display Font', serif;
}
```

🛠️ CSS Variables You Can Use

The project uses CSS variables for consistent styling. You can modify these or use them directly:

Background Colors

- `var(--semantic-bg-base)` - Base background
- `var(--semantic-bg-subtle)` - Subtle background
- `var(--semantic-bg-muted)` - Muted background

Text Colors

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

Border Colors

- `var(--semantic-border-primary)` - Primary borders
- `var(--semantic-border-secondary)` - Secondary borders

Find all variables in: `app/globals.css` and `design-tokens/*.css`

🔄 How to Refresh/Restart

If changes aren't showing:

1. **Hard Refresh Browser:**
   - Windows/Linux: `Ctrl + Shift + R`
   - Mac: `Cmd + Shift + R`

2. **Clear Next.js Cache:**

```
cd /home/ubuntu/code_artifacts/loomOS-styling-playground
rm -rf .next
```

Then the server will automatically rebuild.

1. **Restart Dev Server** (if needed):

```
# Stop the server
pkill -f "next dev"

# Start again
cd /home/ubuntu/code_artifacts/loomOS-styling-playground
npm run dev
```

🎯 Interactive Elements to Style

Cards in Carousel

- Card backgrounds
- Card shadows
- Card border radius
- Card hover effects
- Card sizes and spacing

Top Status Bar

- Background color/gradient
- Icon colors
- Height
- Transparency effects

Global Search

- Input styling
- Placeholder text
- Search results appearance
- Animation effects

Dock

- Background blur/transparency
- Icon colors and sizes
- Position (bottom, left, right)
- Spacing between icons

📱 Testing Responsive Design

The dashboard is responsive. Test different screen sizes:

In Browser DevTools:

1. Open DevTools: `F12` or `Ctrl+Shift+I`
2. Click device toolbar icon or press `Ctrl+Shift+M`
3. Select different devices or set custom dimensions

Common Breakpoints:

- Mobile: 375px - 767px
- Tablet: 768px - 1023px
- Desktop: 1024px+

Responsive styling is in: Tailwind classes (sm:, md:, lg:, xl:)

💡 Pro Tips

1. Use Browser DevTools

- Right-click any element → "Inspect"
- Modify CSS live to test before saving
- Check Console for any errors

2. Keep Changes Small

- Make one change at a time
- Test each change before moving to the next
- Easier to identify what works

3. Use CSS Variables

- More maintainable than hardcoded values
- Easy to create theme variations
- Consistent across components

4. Check Console for Errors

- If something breaks, check browser console
- Common issues: syntax errors, missing imports

5. Experiment Freely!

- This is a playground - break things!
- You can always revert changes via git
- Learn by trying different approaches

🐛 Troubleshooting

Dashboard shows blank screen

- Check browser console for errors
- Ensure dev server is running: `netstat -tuln | grep 3000`
- Try hard refresh: `Ctrl+Shift+R`

Changes not appearing

- Save the file properly
- Wait a few seconds for HMR
- Try hard refresh
- Clear browser cache
- Restart dev server if needed

"Module not found" error

- Check file paths are correct
- Ensure imports match file locations
- Run `npm install` if needed

Port 3000 already in use

```
# Kill existing process
pkill -f "next dev"

# Or use different port
PORT=3001 npm run dev
```

📚 Additional Resources

Documentation Files in This Repo:

- **PLAYGROUND_GUIDE.md** - Comprehensive styling guide
- **QUICK_START.md** - 3-minute quick start
- **DASHBOARD_STRUCTURE.md** - Component architecture reference

External Resources:

- Next.js Docs (https://nextjs.org/docs)
- Tailwind CSS Docs (https://tailwindcss.com/docs)
- CSS Gradients Generator (https://cssgradient.io/)
- Color Palette Generator (https://coolors.co/)

✨ Next Steps

1. **Explore the dashboard** - Click around, open cards, test interactions
2. **Make your first change** - Try changing a card color
3. **Experiment with layouts** - Adjust spacing, sizes, positions
4. **Create custom themes** - Design your own color schemes
5. **Test responsive design** - See how it looks on different devices

---

🎉 Happy Styling!

Your playground is ready. Start experimenting and have fun! Remember:
- Changes are instant with Hot Module Replacement
- You can always undo via git
- This is your safe space to try new ideas

Current Status:
- ✅ Server running on http://localhost:3000
- ✅ Dashboard accessible at http://localhost:3000/dashboard
- ✅ Authentication disabled for easy access
- ✅ Hot reload enabled for instant feedback

Questions or Issues?

Check the troubleshooting section above or refer to the other documentation files in the repository.