

Styling Implementation Guide

What We've Done

1. Enhanced Global CSS (globals.css)

Added comprehensive utility classes for consistent dark theme styling:

Form Elements:

- `form-input-dark` - Dark theme text inputs
- `form-select-dark` - Dark theme dropdowns
- `form-label-dark` - Form labels
- `form-checkbox-dark` - Checkboxes
- `form-radio-dark` - Radio buttons

Buttons:

- `btn-dark-primary` - Blue primary buttons
- `btn-dark-secondary` - Gray secondary buttons
- `btn-dark-success` - Green success buttons
- `btn-dark-danger` - Red danger buttons
- `btn-dark-outline` - Outline style buttons

Cards:

- `card-dark` - Standard cards
- `card-dark-hover` - Cards with hover effects
- `card-dark-highlighted` - Highlighted/selected cards

Badges:

- `badge-dark-success`, `badge-dark-error`, `badge-dark-warning`, `badge-dark-info`, `badge-dark-neutral`

Tables:

- `table-dark` - Complete table styling

Alerts:

- `alert-dark-success`, `alert-dark-error`, `alert-dark-warning`, `alert-dark-info`

2. Matrix Control Page

✓ Already updated with improved styling (completed in previous session)

What Needs to Be Done

The utility classes are ready to use! Now individual components need to be updated to use these classes.

Priority Pages to Update:

1. **Atlas Configuration** (`/atlas-config`)
 - Replace light-themed inputs with `form-input-dark`

- Update buttons to use `btn-dark-*` classes
- Apply `card-dark` to containers

2. **Sports Guide** (`/sports-guide`)

- Update filter controls
- Apply dark theme to search inputs
- Style result cards consistently

3. **Bartender Remote** (`/remote`)

- Enhance control buttons
- Update device selection UI
- Style TV layout visualization

4. **System Logs** (`/logs`)

- Apply `table-dark` to log tables
- Update filter controls
- Style export/download buttons

5. **Device Config Pages**

- DirecTV configuration
- Fire TV configuration
- Global Cache configuration

How to Apply the Updates

Example: Updating a Form

Before:

```
<input
  type="text"
  className="w-full px-4 py-2 border rounded"
  placeholder="Device name"
/>
```

After:

```
<input
  type="text"
  className="form-input-dark"
  placeholder="Device name"
/>
```

Example: Updating a Button

Before:

```
<button className="px-4 py-2 bg-blue-500 text-white rounded">
  Save
</button>
```

After:

```
<button className="btn-dark-primary">
  Save
</button>
```

Example: Updating a Card

Before:

```
<div className="bg-white rounded-lg border p-6">
  <h3 className="text-lg font-semibold">Title</h3>
  <p>Content</p>
</div>
```

After:

```
<div className="card-dark p-6">
  <h3 className="text-xl font-semibold text-slate-100 mb-4">Title</h3>
  <p className="text-slate-300">Content</p>
</div>
```

Testing Each Update

After updating a component:

1. Verify text is readable
2. Check contrast ratios
3. Test hover states
4. Confirm focus indicators work
5. Test on mobile devices

Benefits

- **Consistency:** All pages look cohesive
- **Readability:** Better contrast for accessibility
- **Professional:** Modern, polished appearance
- **Maintainable:** Utility classes make updates easy
- **Accessible:** WCAG AA compliant contrast ratios

Next Steps

You can now:

1. Continue using the app with improved Matrix Control page styling
2. Update additional pages as needed using the new utility classes
3. Reference `SITE_WIDE_UI_IMPROVEMENTS.md` for detailed documentation
4. Apply the patterns gradually to each component

All changes have been committed and pushed to GitHub!