

Design System

MUI component library + dark theme

A portfolio case study about extending Material UI (MUI) into reusable, predictable and scalable enterprise components across light and dark experiences.

MUI components

States & variants

Light + dark themes

Documentation

Extending MUI for enterprise product needs

The work used Material UI (MUI) as the component foundation, then adapted component behavior, variants, styling and theme treatment for ConnectWise product experiences.

Context

Enterprise SaaS workflows need a shared UI language across dense, data-rich product areas.

My contribution

MUI component design, variants, states, interaction logic, dark-theme treatment, accessibility considerations and usage guidance.

Design intent

Use MUI as a robust base while creating consistent patterns that fit product requirements and scale across teams.

A library is a starting point, not the finished system

MUI provides reusable primitives, but enterprise products still need product-specific decisions around behavior, density, theming, accessibility and documentation.

01

Product consistency

Create predictable patterns across different workflows and teams.

02

MUI adaptation

Configure and extend components without creating unnecessary one-off patterns.

03

Theme parity

Ensure light and dark experiences preserve hierarchy, states and usability.

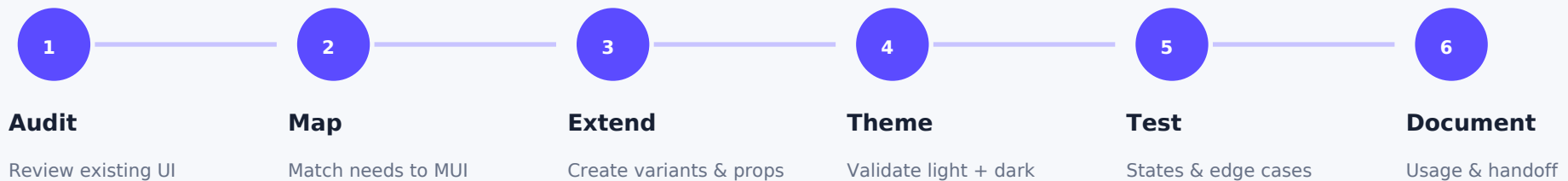
04

Handoff clarity

Make component behavior explicit for designers and engineers.

From product audit to reusable MUI patterns

A portfolio-friendly view of the end-to-end component process.



DESIGN PRINCIPLE

Prefer a shared MUI-based pattern when it solves the product need; extend deliberately when enterprise workflows require additional behavior or presentation.

Theme decisions connect the component family

Tokens and theme rules help MUI components feel like one product system instead of a collection of controls.

Palette

Semantic roles for actions, surfaces, text and status.



Typography

Consistent hierarchy across labels, body content and data.

Spacing & shape

Predictable density, radius and layout rhythm.

Use the actual ConnectWise/MUI theme token names only if they are approved for public portfolio use.

MUI primitive -> product-ready component

The component story should show what was inherited from MUI and what was configured or extended for product needs.

Example / MUI TextField

Email address

name@company.com

Label

Container

Value

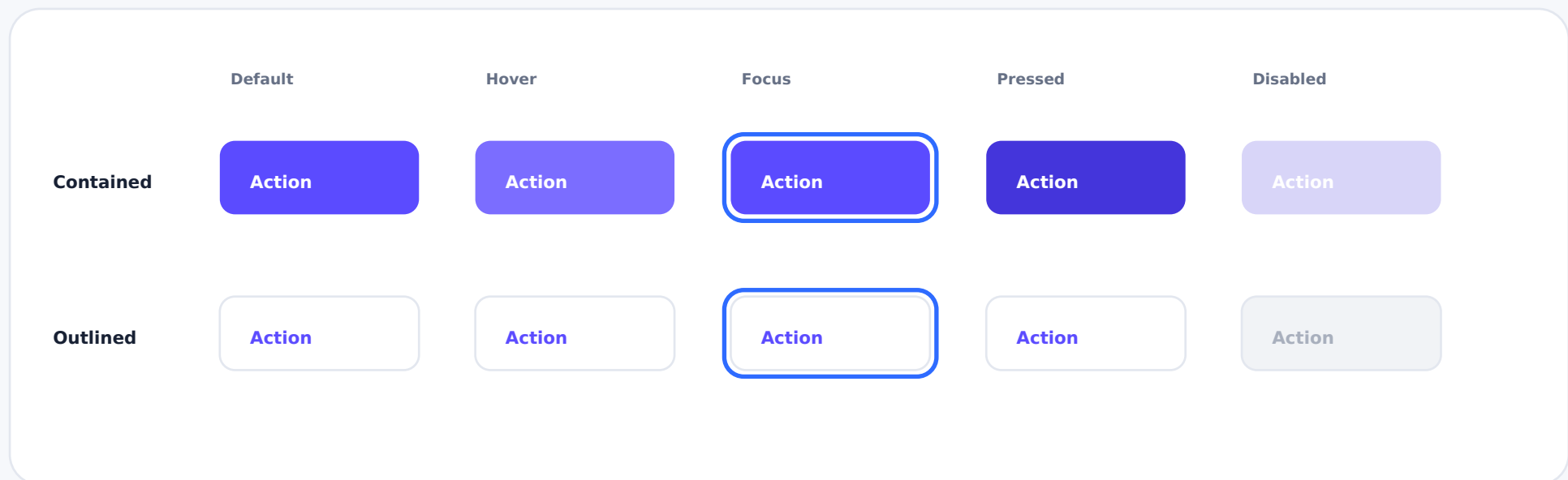
Focus state

Extension checklist

- ✓ MUI base component
- ✓ Product variant
- ✓ State behavior
- ✓ Theme tokens
- ✓ Accessibility
- ✓ Responsive context

Define behavior across the component lifecycle

MUI components were treated as configurable systems with explicit states and variants.



EXAMPLE PROPERTY MODEL

Variant = contained / outlined | Size = small / medium / large | State = default / hover / focus / pressed / disabled | Icon = optional.

The same component language, rebalanced for dark surfaces

Dark theme was treated as a system state, not a simple color inversion. Components were checked for surface hierarchy, contrast, focus visibility, disabled states and semantic feedback.

MUI component examples / dark theme

Search

Search tickets

Primary action

Secondary

Disabled

Theme checks

- **Surface hierarchy**
Separate background, paper and elevated surfaces.

- **Contrast**
Preserve readable text and interactive states.

- **Semantic color**
Rebalance success, warning and error against dark surfaces.

- **Focus & disabled**
Keep keyboard focus visible and disabled states understandable.

Reusable means understandable

Visual specifications were paired with interaction and usage guidance.

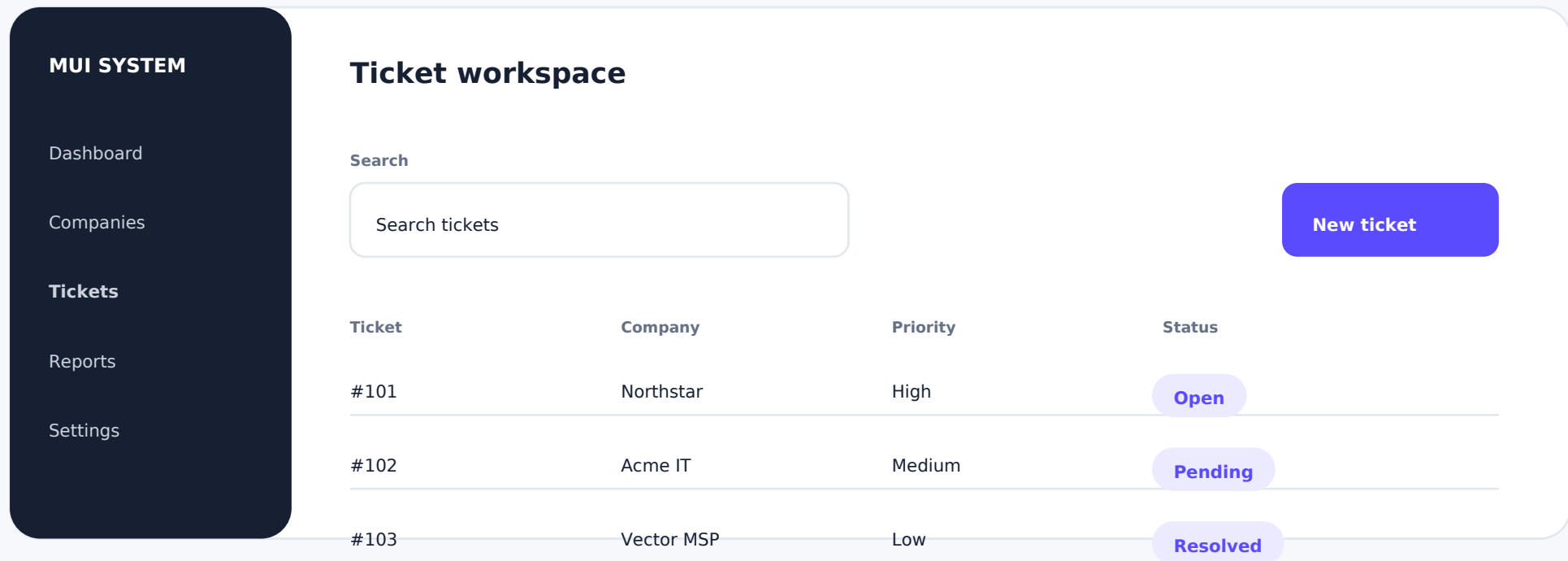
- **Focus**
Visible keyboard focus treatment.
- **Contrast**
Readable text and states in both themes.
- **Meaning**
Status is not communicated by color alone.
- **Content**
Labels and error messages remain clear.

Documentation template

MUI source	Base component / primitive.
When to use	Intended product scenario.
Variants	Supported configurations.
Behavior	States and interactions.
Theme	Light and dark treatment.
Do / Don't	Correct and incorrect usage.

Components have to survive real enterprise workflows

The final check is how the MUI-based components behave together inside dense product scenarios.



MUI provided the foundation. The design system created the

The work focused on turning reusable MUI primitives into consistent enterprise patterns across states, workflows and both light and dark themes.

Consistency

Shared MUI-based patterns

Efficiency

Less repeated component design

Theme parity

Coherent light and dark experiences

Alignment

Clearer design-to-development intent

Portfolio note

Replace these illustrative controls with your actual approved ConnectWise component screenshots for the final public case study.