

Implement High-Fidelity Designs with MUI and React

Bridge the gap between design and development by turning detailed mockups into production-ready React components using Material-UI.

Web & Mobile Design

Instructor: Morgan Blake • 5 sections • 56 lessons • 21.25h total

[View this course online](#)

[Browse all courses](#)

About this course

Designers hand off pixel-perfect mockups, and developers are expected to match them exactly. In practice, that rarely happens smoothly. Colors drift, spacing breaks, responsive behavior diverges, and the final product looks 'close enough' instead of right. This course fixes that. You'll work through the full translation process, from reading Figma files to writing MUI components that match every detail.

We start with project setup and MUI fundamentals, then move into reading design files, mapping tokens to themes, customizing components, handling animations and accessibility, and wrapping up with a full SaaS dashboard capstone built from a real Figma design. Lessons include step-by-step coding sessions, real case studies from production apps, and problem-solving exercises that mirror what you'll face on the job.

Course curriculum

Getting Started with React and Material-UI

- **Set Up a Modern React Project with Vite and Node.js (22m)**
Scaffold a fast React project using Vite. Walk through folder structure, hot module replacement, and why Vite replaced Create React App for most teams in 2026.
- **Install and Configure Material-UI (MUI v6+) (15m)**
Add MUI to your project with npm, configure the peer dependencies, and verify everything renders correctly in the browser.
- **Understand the Component Library Structure (20m)**
Explore MUI's package structure: core components, icons, lab (experimental) components, and the system utilities. Know what lives where and when to reach for each.
- **Explore the MUI Theme Object (28m)**
Break down the theme object piece by piece. We'll cover palette, typography, spacing, shape, and breakpoints so you understand how MUI controls visual output.
- **Build Your First Styled Button with MUI (14m)**
Create a button using MUI's Button component, apply variant and color props, then customize it with the sx prop for a specific design look.
- **Create Cards and Content Blocks (18m)**
Build a product card component from scratch. Covers Card, CardContent, CardMedia, and CardActions with practical styling patterns.
- **Lay Out Pages with Grid and Container (24m)**
Use MUI's Grid v2 and Container components to create responsive page layouts. We'll build a three-column layout that collapses cleanly on mobile.
- **Project Setup Review and Configuration Tips (12m)**
Review what we've built so far. Cover common setup issues, environment configuration, and editor settings that speed up MUI development.

Bridging Design Files to Code

- **Read and Interpret High-Fidelity Designs from Figma (25m)**
Open a real Figma file and learn how to inspect layers, measure spacing, read typography styles, and extract color values. No guessing, just systematic reading.
- **Identify Design Tokens: Colors, Fonts, Radii, and Shadows (20m)**
Pull every reusable value from a design file. We'll create a structured token list covering the full color palette, font stack, border radii, and elevation shadows.
- **Map Design Tokens to a Custom MUI Theme (30m)**
Take those extracted tokens and build a complete MUI theme. Step-by-step translation of every color, font weight, and spacing value into theme configuration.
- **Use the MUI sx Prop for Inline Customization (18m)**
Apply one-off styles that match design specs using the sx prop. We'll cover shorthand syntax, theme-aware values, and when sx makes more sense than styled().

- **Work with the styled() API for Reusable Custom Components (22m)**
Create styled wrapper components for patterns that repeat across the design. Learn how styled() differs from sx and when to use each approach.
- **Translate Spacing and Grid Systems for Responsive Layouts (24m)**
Match the designer's spacing scale to MUI's system. We'll handle margin, padding, gap, and grid column behavior across breakpoints.
- **Match Typography Across Devices and Screen Sizes (16m)**
Implement responsive font sizes, line heights, and letter spacing that match the design at every breakpoint. Includes scaling strategies for fluid typography.
- **Practical Exercise: Convert a Hero Section from Mockup to Code (Part 1) (32m)**
Take a complete hero section design and start translating it. We'll extract tokens, set up the layout structure, and build the headline and subtext components.
- **Practical Exercise: Convert a Hero Section from Mockup to Code (Part 2) (28m)**
Finish the hero section with the CTA buttons, background image treatment, responsive behavior, and final pixel-check against the Figma file.
- **Review the Translated Component Against the Original Design (15m)**
Overlay the code output on the design file. Find discrepancies, fix them, and build a habit of doing this review on every component you ship.
- **Common Pitfalls When Bridging Designs to Code (14m)**
Avoid the mistakes that cause the most rework: ignoring border-box behavior, missing auto margins, wrong font loading, and overlooking shadow layering.

Deep Component Customization and Theming

- **Override Default MUI Component Styles with theme.components (26m)**
Use the theme's components key to change default styles globally. We'll restyle buttons, inputs, and cards so they match a custom design system by default.
- **Apply CSS Overrides for Global Style Adjustments (16m)**
Write CSS overrides that target MUI's internal class names. Covers styleOverrides, defaultProps, and handling specificity conflicts.
- **Build a Custom Data Table Component (35m)**
Create a compound data table with sorting, pagination, and row selection using MUI's Table components. Match the design spec for header, body, and footer styles.
- **Build a Multi-Step Form Component (30m)**
Assemble a multi-step form with MUI's Stepper, TextField, Select, and Button components. Handle step navigation, validation, and state management.
- **Build a Navigation Drawer Component (28m)**
Create a responsive navigation drawer that collapses to a bottom sheet on mobile. Uses MUI Drawer, List, and ListItem with custom styling.
- **Work with CssVarsProvider for Light and Dark Mode (22m)**
Set up CssVarsProvider to support automatic and manual light/dark mode switching. Define color schemes and test the toggle behavior.

- **Implement Dynamic Theming with Custom Palettes (20m)**
Go beyond light and dark. Build a theme switcher that lets users pick from multiple color palettes, stored in localStorage and applied at runtime.
- **Integrate Custom Icons and Illustrations (18m)**
Add SVG icons and illustrations that aren't in MUI's default icon set. Register them as components, apply consistent sizing, and handle accessibility.
- **Add Custom Typography Beyond MUI Defaults (15m)**
Load and configure custom font families. Extend the theme's typography variants so every heading and body style matches the design system.
- **Handle Design-System Variants and Custom States (24m)**
Create custom component variants for buttons, chips, and alerts. Define hover, active, disabled, and focus states that match the design specification exactly.

Advanced Interactions, Animations, and Accessibility

- **Add Motion with Framer Motion for UI Feedback (28m)**
Install Framer Motion and add entrance animations, hover effects, and page transitions to your MUI components. Practical patterns, not decoration.
- **Use MUI's Built-In Transitions and Collapse Effects (18m)**
Apply MUI's Collapse, Fade, Grow, Slide, and Zoom transitions. Know when to use each and how to customize duration and easing.
- **Build a Dashboard Panel Layout (32m)**
Create a multi-panel dashboard layout with draggable cards, collapsible sections, and responsive grid behavior. Matches a real SaaS dashboard design.
- **Build a Mega-Menu Navigation Component (26m)**
Implement a mega-menu with MUI Popper, MenuList, and custom hover behavior. Includes keyboard navigation support and mobile fallback.
- **Build a Responsive Sidebar with Nested Navigation (24m)**
Create a sidebar that works as a persistent drawer on desktop and a temporary drawer on mobile. Handles nested menu items and active state tracking.
- **Ensure WCAG 2.2 Compliance: Focus Management (20m)**
Implement proper focus trapping in modals and drawers, manage focus on route changes, and ensure keyboard users can navigate every interactive element.
- **Ensure WCAG 2.2 Compliance: ARIA Attributes (18m)**
Add correct ARIA labels, roles, and live regions to MUI components. Covers custom components where MUI doesn't handle ARIA automatically.
- **Ensure WCAG 2.2 Compliance: Color Contrast and Visual Clarity (16m)**
Test and fix color contrast issues in your theme. Handle text on images, disabled states, and focus indicators so they meet WCAG 2.2 AA standards.
- **Manage Form State with React Hook Form and MUI (28m)**
Connect React Hook Form to MUI's TextField, Select, Checkbox, and Radio components. Covers Controller usage, default values, and field-level validation.

- **Validate Forms with Error Messages and Feedback (20m)**
Display validation errors using MUI's FormHelperText and error props. Build patterns for inline errors, summary errors, and success confirmation.
- **Performance: Lazy Load Routes and Heavy Components (16m)**
Use React.lazy and Suspense to defer loading of large page components. Reduce initial bundle size without visible delay for users.
- **Performance: Tree-Shake MUI Imports Correctly (12m)**
Import only what you need from MUI to keep bundle size small. Covers named imports, Babel plugins, and verifying your build output.
- **Performance: Memoization Strategies with React.memo and useMemo (22m)**
Prevent unnecessary re-renders in complex MUI layouts. Practical examples of when memo helps, when it doesn't, and how to profile with React DevTools.

Production Workflow and Capstone Project

- **Set Up a Component Library with Storybook (30m)**
Install Storybook, configure it for MUI components, and write stories that showcase every variant and state of your custom components.
- **Document Components for Design-Dev Handoff (18m)**
Write clear component documentation with props tables, usage examples, and design notes. Make it easy for designers and other devs to use your components.
- **Implement Visual Regression Testing in CI/CD (24m)**
Add Chromatic or Percy to your pipeline to catch visual changes automatically. Set baselines, review diffs, and prevent unintended style regressions.
- **Compare Design Specs Against Rendered Output (20m)**
Use tools and techniques to overlay rendered components on the original design. Automate checks where possible and build a manual review habit.
- **Integrate Figma Tokens into Your MUI Theme Pipeline (26m)**
Pull design tokens directly from Figma Tokens (or Figma Variables) and sync them to your MUI theme. Automate the pipeline so changes propagate without manual updates.
- **Manage Design Tokens with Style Dictionary (22m)**
Set up Style Dictionary to transform and distribute design tokens across platforms. Generate MUI theme objects, CSS variables, and documentation from one source.
- **Capstone: Plan the SaaS Dashboard Structure (20m)**
Analyze the full Figma design file for the capstone project. Break it into pages, sections, and components. Plan the folder structure and component hierarchy.
- **Capstone: Build the Dashboard Layout and Navigation (35m)**
Implement the main layout shell with the sidebar, top bar, and content area. Set up routing and connect navigation to page components.
- **Capstone: Implement Core Dashboard Widgets (40m)**
Build stat cards, chart containers, activity feeds, and user tables. Each widget matches the Figma spec for size, spacing, and visual treatment.

- **Capstone: Add Charts, Data Views, and Interactive Elements (38m)**
Integrate a charting library with MUI-styled wrappers. Build filter controls, sort toggles, and pagination for data-heavy views.
- **Capstone: Polish Responsive Behavior and Animations (30m)**
Test every page at mobile, tablet, and desktop widths. Add entrance animations, loading states, and empty-state screens for a complete experience.
- **Capstone: Deploy the Finished Dashboard (22m)**
Build for production, configure environment variables, and deploy to a hosting platform. Verify performance scores and fix any build warnings.
- **Package Your Work for a Portfolio (16m)**
Prepare screenshots, a live demo link, and a written case study that explains your process. Structure it so hiring managers and clients can evaluate your work quickly.
- **Best Practices for Ongoing Design-Development Collaboration (18m)**
Wrap up with workflow habits that keep design and code in sync over time. Covers component libraries, token governance, and communication patterns between teams.

Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **[Flight Attendant Preparation: What to Study Before You Apply](#)**

[Visit the blog](#) [All courses](#) [This course](#)