

UI Component Design — Workbook

This workbook turns the course into a build. You will pick a small set of real components, a button, a text field, a modal, a navigation bar, and a card, and design each one with its full state matrix, tokens, and accessibility checks. Work through the four sections in order; by the end you will have a documented mini component library and a developer handoff spec you can put in a portfolio.

From Screens to Components

Reframe your work as components, set up an 8-point spacing system, and lay out a state matrix.

Worksheet: Component Anatomy Brief

Choose one component you will design through this whole workbook (a button is the recommended first). Fill in each field. The one-sentence job is your contract: every later decision must serve it.

Component name (e.g. Primary Button)

One-sentence job of this component

Variants (meaningful alternatives, e.g. primary / secondary / destructive)

Sizes (if any, e.g. small / medium / large)

States that genuinely apply (default / hover / focus / active / disabled / loading / error)

States that do NOT apply (and why)

Equivalent coded prop names a developer would expect (e.g. variant, size, disabled, loading)

Exercise: Spot the Copy-Paste Components

Open any app or website you use daily. Find three UI elements that repeat across screens. For each, note evidence that it should be a single reusable component rather than copied shapes.

- Which element repeats most across screens, and how many places did you spot it?

- Did any two instances differ slightly (padding, colour, radius) in a way that looks like drift, not intent?

- What would you name each as a component using a general-to-specific slash convention?

- Which states did you observe each element in (hover, disabled, etc.), and which were missing or broken?

Worksheet: 8-Point Spacing Scale and Grid Setup

Define your spacing scale and layout grid before drawing anything. Use multiples of 8 with 4 as a half-step. Record the token name you will use for each value.

space-1 value (suggest 4px) and what it is used for

space-2 value (suggest 8px) and use

space-3 value (suggest 16px) and use

space-4 value (suggest 24px) and use

space-5 value (suggest 32px) and use

space-6 value (suggest 48px) and use

Desktop layout grid (columns / gutter / margin, e.g. 12 / 24px / 80px)

Mobile layout grid (columns / gutter / margin)

Figma nudge settings (small nudge / large nudge)

Checklist: Foundations Ready Check

- I wrote a one-sentence job for my component
- I listed which states apply and which do not
- I chose a naming convention (PascalCase components, kebab-case tokens) and wrote it down
- I defined an 8-point spacing scale with token names
- I set up a 12-column desktop grid and a mobile grid in Figma
- I set Figma nudge amounts so arrow keys move by 8

Design Tokens and Accessible Foundations

Build a layered token set and prove your colours, targets, and focus states meet accessibility minimums.

Worksheet: Three-Tier Token Map

Map your colours through primitive, semantic, and (optionally) component layers. Fill the primitive hex values and the semantic names that point at them. Leave the measured contrast cells for the contrast worksheet below.

Primitive: color-brand-500 hex

Primitive: color-gray-900 (text) hex

Primitive: color-gray-500 (muted) hex

Primitive: color-danger-500 hex

Semantic: color-action-primary points at which primitive

Semantic: color-text-default points at which primitive

Semantic: color-feedback-danger points at which primitive

Type scale tokens (text-xs / sm / base / lg / xl values)

Radius tokens (radius-sm / md / full values)

Exercise: Run a Real Contrast Audit

Using the WebAIM Contrast Checker or a Figma plugin (Stark / Contrast), measure each text-on-background pairing in your design. Record the ratio and whether it passes the relevant WCAG AA threshold (4.5:1 normal text, 3:1 large text and UI components).

- What is the measured ratio of your default body text on its background, and does it clear 4.5:1?

- What is the ratio of your placeholder or muted text, and does it pass (most light grey placeholders fail)?

- What is the ratio of your primary button label on its fill?

- Does any colour-coded meaning (error, success, status) lack a second non-colour cue, and how will you fix it?

Worksheet: Touch Target and Focus State Plan

For each interactive component, define its hit area (not just visual size) and its focus ring. Aim for 44 to 48px targets and a visible focus ring with at least 3:1 contrast.

Component name

Visible size (width x height)

Hit area size (target 44 to 48px; note added padding if icon is smaller)

Spacing to nearest adjacent target (target at least 8px)

Focus ring thickness (e.g. 2 to 3px)

Focus ring colour token and measured contrast vs background

Focus ring offset (e.g. 2px)

Checklist: Accessible Foundations Check

- [] Every colour decision uses a semantic token, not a raw hex, in my components
- [] Body text clears 4.5:1 and large text / UI elements clear 3:1 (measured, not guessed)
- [] No meaning relies on colour alone; each has a second cue (icon, shape, or text)
- [] I ran my palette through a colour-blindness simulator
- [] Every interactive target is at least 44 to 48px in hit area
- [] Every interactive component has a visible focus state at 3:1+ contrast

Building the Core Components

Design buttons, form fields, modals, and navigation with complete state matrices and correct behaviour.

Worksheet: Button State Matrix Builder

Complete the state matrix for your button. For each variant and state, describe the exact visual treatment (fill, border, text colour token, opacity, ring). This becomes your Figma component set.

Primary, default treatment

Primary, hover (e.g. 10 to 15 percent darker fill)

Primary, focus (ring spec)

Primary, active / pressed

Primary, disabled (muted token / opacity)

Primary, loading (spinner + fixed width note)

Secondary, treatment across the same six states

Destructive, treatment across the same six states

Padding (vertical / horizontal) and radius token used

Worksheet: Form Field Specification

Design one text field fully. Define each anatomical part and each state. Confirm the label is persistent and the error state uses more than colour.

Label text and position (above the field recommended)

Input height and internal padding

Placeholder text (a hint only, never the label)

Helper text (persistent guidance, if any)

Focus state treatment (border / ring at 3:1+ contrast)

Error state treatment (danger colour + icon + message text)

Example error message (specific and actionable, e.g. 'Enter a valid email, for example name@example.com')

Validation timing (recommended: on blur, clear on correction)

Exercise: Modal Behaviour Walkthrough

Design a modal, then write out its keyboard and focus behaviour step by step. If you cannot answer every prompt, the modal spec is incomplete.

- When the modal opens, where exactly does keyboard focus land?
 - How is focus trapped so Tab cannot reach the page behind?
 - What closes the modal (Escape, scrim click, X button) and which closures are disabled for critical confirmations?
 - On close, which element receives focus, and is it the element that opened the modal?
-

Checklist: Navigation Component Check

- The current page / selected tab has an unmistakable active state that does not rely on colour alone
- The active indicator clears 3:1 contrast against its background
- I designed both the desktop nav and its mobile form (hamburger drawer or bottom bar) and chose the breakpoint
- The menu toggle has its own focus and expanded states
- Keyboard users can open, traverse, and close the navigation
- Top-level mobile destinations are limited to three to five items

Cards, Documentation, and Handoff

Design a flexible card, document the library, and produce a handoff spec with a final pre-ship review.

Worksheet: Card Structure and Content Plan

Design one card around its one-sentence job. Define its structure, tokens, and how it handles real, messy content.

Card one-sentence job

Vertical structure (media / title / supporting text / metadata / actions)

Image aspect ratio (fixed, so the grid stays aligned)

Internal padding token and radius token

Elevation cue (shadow OR 1px border, pick one as the system convention)

Interactive states (hover and focus, if the card is clickable)

Long-title handling (truncate with ellipsis OR wrap to N lines)

If the whole card is clickable AND contains a button: how the two click targets coexist

Worksheet: Component Documentation Page

For each finished component, write its documentation entry. Repeat this worksheet per component. Keep dos and don'ts concrete.

Component name and one-sentence job

When to use each variant

Do (a correct-usage rule)

Do (a second correct-usage rule)

Don't (a misuse to avoid)

Don't (a second misuse to avoid)

Is the full state matrix shown on the documentation frame? (yes/no)

Published to the shared Figma library? (yes/no)

Exercise: Pre-Ship Self-Review

Run this review on every component before declaring it done. Note any failures and fix them before moving on. Relying on memory guarantees you miss the focus state on the component a keyboard user needs most.

- Is every applicable interaction state designed (default, hover, focus, active, disabled, loading, error)?
 - Does every text element clear 4.5:1 and every UI element / large text clear 3:1 (measured)?
 - Is every touch target at least 44 to 48px and every focus state visible?
 - Does spacing follow the 8-point scale and do components reference tokens instead of hard-coded values?
-

Checklist: Developer Handoff Complete Check

- Each component lists the named tokens it uses (colour, spacing, type, radius), not raw values
- All applicable states are shown with notes
- Behaviour is annotated: click, submit, open/close, focus movement, and keyboard keys
- Edge cases are documented: empty state, loading, long text, and error handling
- Accessibility notes included: label associations, ARIA roles/states, focus order, target sizes
- Responsive rules and breakpoints are documented
- I walked a developer (or a peer) through the spec and they had no open questions

Your Action Plan

1. Pick five real components to design this week: button, text field, modal, navigation bar, and card.
2. Set up your foundations in Figma: an 8-point spacing scale, a 12-column desktop grid and mobile grid, and nudge settings.
3. Build a three-tier token set (primitive, semantic, optional component) for colour, type, and radius using Figma Variables.
4. Run a contrast audit on every text and UI pairing and fix anything below 4.5:1 (text) or 3:1 (large text and UI).
5. Design the button first as a full variant-by-state matrix using component properties, then reuse the discipline on the rest.
6. Design the text field with a persistent label and an error state that uses colour, an icon, and a message together.
7. Design the modal and navigation with explicit focus, keyboard, and responsive behaviour written down.
8. Design the card around a one-sentence job and test it with short and long content and a fixed image aspect ratio.
9. Write a documentation page per component with dos, don'ts, and a visible state matrix, then publish to a shared library.
10. Produce a handoff spec, run the pre-ship self-review, and walk a developer or peer through it until they have no questions.

