

Production-Ready UI System for Figma

A field guide to building responsive interfaces,
documenting intent, and handing designs to developers
without the usual guesswork.



This is a production system, not a gallery of pretty screens.

Use it when a design must survive real content, multiple viewports, accessibility requirements, and an actual developer handoff.

1

Design coherently

Start from shared tokens and responsive primitives instead of recreating foundations in every file.

2

Expose intent

Document behavior, states, edge cases, assets, and accessibility close to the interface.

3

Verify the build

Use the same acceptance criteria during handoff and design QA after implementation.

Use it for

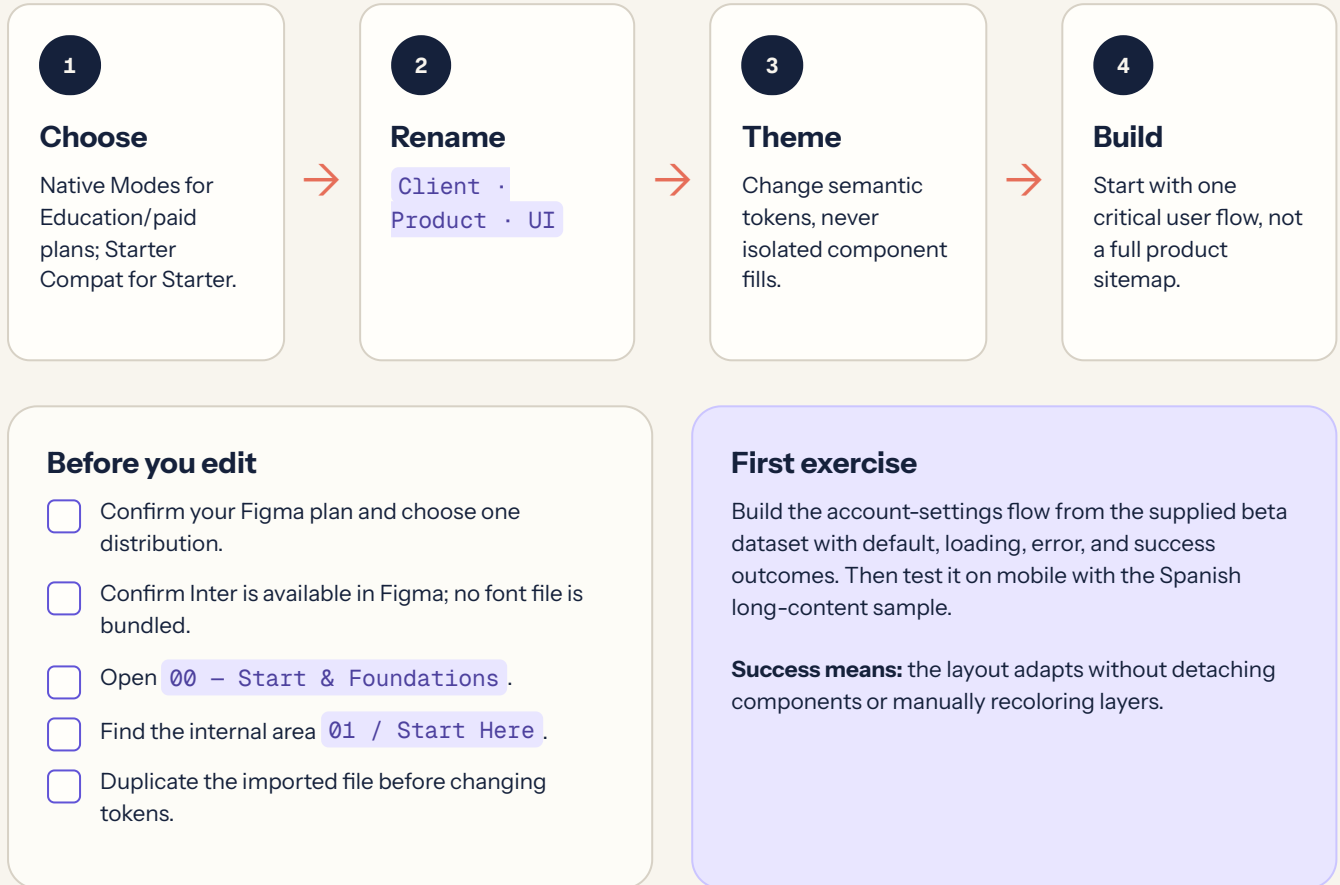
- ☐ SaaS and dashboard interfaces
- ☐ Responsive websites and web apps
- ☐ Agency projects with developer handoff
- ☐ Small product teams without a mature system

Do not use it as

- ☐ A finished visual identity
- ☐ A substitute for user research
- ☐ An automatic design-to-code exporter
- ☐ A guarantee that every implementation decision is solved

The rule: customize the visual language; preserve the structural logic.

Choose the correct file, duplicate it, and begin with a real flow.



Distribution rule: Native uses real theme and viewport modes. Starter Compat uses fixed Light/Dark structures and exposed properties. Do not mix both files in one project.

Two physical pages contain every production area.

00	Start & Foundations	The first physical Figma page.
A	Cover + Start Here	Version, compatibility, setup, license and first exercise.
B	Color & Themes	Foundation and semantic colors for Light and Dark.
C	Type, Metrics, Responsive & Content	Typography, spacing, grids, breakpoints and stress rules.
01	Components, Patterns & Templates	The second physical Figma page.
D	35 Components + 10 Patterns	Reusable sets and functional composition examples.
E	4 Template Families	24 theme and viewport combinations.
F	Handoff, QA, Playground & Changelog	Internal delivery, testing, practice and release areas.

Navigation habit: components live in one place. Project screens should use instances; never move the main component set into a client flow.

Change meaning at the semantic layer, not color values one component at a time.

background/brand · Light
#584ED8

text/primary · Light
#101828

action/danger/background
#D92D20

Token chain

`foundation.color.danger.600`

↓

`component.action.danger.background`

↓

`--component-action-danger-background`

Primitives describe values. Semantic tokens describe purpose. Component tokens describe a controlled exception.

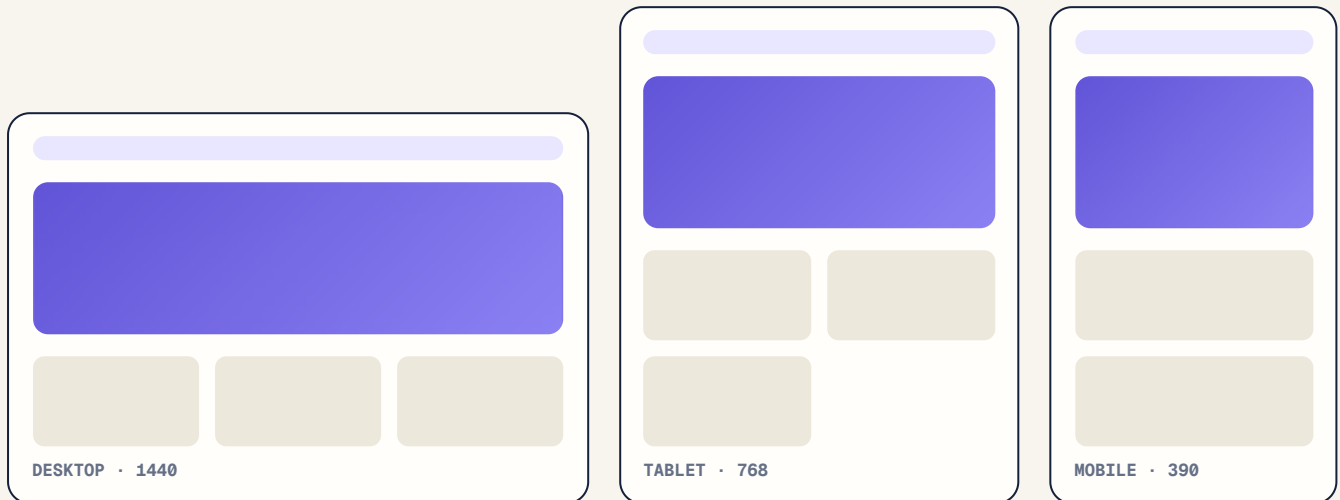
Recommended edit order

1. Replace brand primitives.
2. Review semantic aliases.
3. Check contrast in both themes.
4. Change typography variables and styles.
5. Review only necessary component exceptions.

Keep Figma and code speaking the same language

FIGMA VARIABLE	CSS TOKEN	PURPOSE
control/radius	<code>--component-control-radius</code>	Shared control corner radius.
field/gap	<code>--component-field-gap</code>	Field label/control spacing.
action/danger/background	<code>--component-action-danger-background</code>	Destructive action background.
drawer/width/md	<code>--component-drawer-width-md</code>	Medium side drawer width.

Design the rule between viewports, not three unrelated screenshots.



Resize

A fluid element keeps its structure and changes available width.

Reflow

Columns become rows, or a grid changes its column count.

Wrap

Items continue on another line when space becomes insufficient.

Hide

Secondary information is removed only when it remains available elsewhere.

Swap

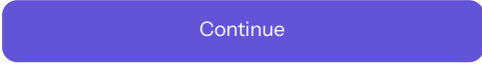
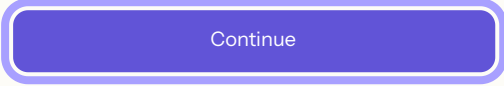
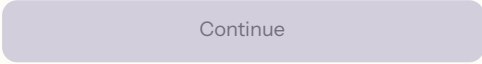
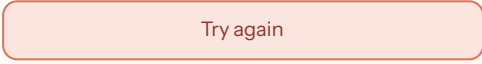
A desktop pattern changes to a genuinely different mobile component.

Scroll

Content remains intact inside an intentional overflow container.

Annotate the behavior: “three columns at 1024 px and wider; two between 640–1023; one below 640” is more useful than three unlabeled frames.

Every interactive component needs a complete behavioral story.

DEFAULT		Available, not yet interacted with.
FOCUS		Visible keyboard focus; never remove it without replacement.
DISABLED		Unavailable and paired with an explanation when needed.
ERROR		Explains what happened and how to recover.

Property naming

- ☐ Hierarchy=Primary
- ☐ State=Default
- ☐ Size=Medium
- ☐ Icon start=True
- ☐ Icon end=False

Avoid

- ☐ Variant 12 or visual-only names.
- ☐ Boolean properties with unclear meaning.
- ☐ Hundreds of combinations nobody uses.
- ☐ Different component structures for minor visual changes.
- ☐ Detached instances in finished flows.

Design every outcome: default, hover, focus, active, disabled, loading, empty, success, and error — when they genuinely apply.

Test the uncomfortable screen before a user finds it for you.

Content stress test

- ☐ 200% expected text length.
- ☐ Zero results and first-use state.
- ☐ Broken or missing image.
- ☐ Maximum table rows.
- ☐ Slow loading and partial data.
- ☐ Long email, URL, price, and filename.

Accessibility annotation

- ☐ Element role: link, button, heading, list.
- ☐ Accessible name and error association.
- ☐ Keyboard and focus behavior.
- ☐ Reading and focus order.
- ☐ Contrast and non-color cues.
- ☐ Reduced-motion alternative.

Use this annotation pattern

ROLE · Button

NAME · “Save project”

KEYBOARD · Enter or Space activates

BUSY · Preserve label; show progress indicator; prevent duplicate submission

SUCCESS · Return focus to heading; announce confirmation

Annotations communicate intent; they do not replace implementation testing with a keyboard, screen reader, zoom, or real content.

Mark a design ready only when the questions have answers.

Before Ready for Dev

- ☐ Scope and user flow are approved.
- ☐ Responsive rules are documented.
- ☐ Relevant interaction states exist.
- ☐ Tokens replace arbitrary values.
- ☐ Assets have locations and export formats.
- ☐ Accessibility behavior is annotated.
- ☐ Known limitations are visible.

Definition block

Purpose

What user problem does this flow solve?

Behavior

What changes, persists, scrolls, or receives focus?

Data

What can be empty, slow, long, or unavailable?

Acceptance

What must be true for this to be approved?

Recommended handoff sequence

01

Walk through the user flow.

02

Review behavior and edge cases.

03

Map tokens and components.

04

Resolve open implementation questions.

05

Agree on design-QA checkpoints.

A Figma link is not a handoff. The handoff is a shared understanding of behavior, constraints, and approval criteria.

Compare the build against intent, not only pixel position.

AREA	CHECK	EVIDENCE
Structure	Correct component, hierarchy, and content order.	Screenshot + DOM review
Tokens	Color, spacing, radius, type, and elevation use system values.	Inspector
Responsive	Layout behaves between reference frames, not only at them.	Viewport sweep
Interaction	Hover, focus, active, disabled, loading, success, and error.	Recorded flow
Content	Long, empty, localized, slow, and broken-data scenarios.	Fixtures
Accessibility	Keyboard, names, roles, order, contrast, zoom, and motion.	Manual + automated tests
Assets	Correct format, crop, sharpness, and alternative text.	Network + visual check

P0 • Blocker

Broken import, data loss or a core workflow impossible.
Fix before testing continues.

P1 • Major

Wrong tokens/states, major access failure or rebuilding
a core task. Fix before approval.

P2 • Moderate

Confusing instruction or inconsistency with a
documented workaround. Track and decide.

P3 • Polish

Optional refinement with no measurable task or
consistency impact.

Know what is included, what depends on your plan, and how updates work.

Included

- ☐ Two editable Figma distributions.
- ☐ Inter referenced in Figma; no font file bundled.
- ☐ DTCG design-token JSON.
- ☐ CSS custom-properties file.
- ☐ Quick Start Guide.
- ☐ Component and design-QA checklists.
- ☐ Compatibility, mapping, notices and known limitations.

Not included

- ☐ Figma, hosting, or third-party subscriptions.
- ☐ A finished brand identity.
- ☐ Production application code.
- ☐ Custom implementation consulting.
- ☐ A guarantee against future Figma changes.
- ☐ Rights to redistribute the source as a competing kit.

When a new version is released

1. Read packaged `CHANGELOG.md` and `KNOWN-LIMITATIONS.md`.
2. Duplicate the new source as a separate file.
3. Compare token and component changes against your customized version.
4. Adopt relevant changes deliberately; never overwrite a live client library blindly.
5. Run the smoke-test flow again.

Support request format: product version, Figma plan, duplicated minimal example, expected result, actual result, and screenshot or short recording.

RC 0.12.0 is for closed-beta evaluation. Its license is a draft and is not approval for public redistribution or paid resale.

Design the happy path. Then design everything that can disturb it.

Build with tokens. Test with uncomfortable content.
Explain behavior. Hand off together. Verify the
implementation.

- 01 · Define the flow
- 02 · Apply foundations
- 03 · Compose with instances
- 04 · Stress-test content and states
- 05 · Annotate intent
- 06 · Mark Ready for Dev
- 07 · Run design QA