

CASE STUDY

Colooors

Complete Color Toolkit for Modern Designers

Turning eight scattered color tools into one fast, delightful workflow for designers and developers.

CLIENT

Colooors

INDUSTRY

Design Tools, SaaS

SERVICES

UX Strategy, UI Design, Design System

PROJECT SNAPSHOT

A single toolkit, not eight separate apps

Colooors already had the tools designers needed: palette generation, gradients, an accessibility checker, and a full color library. Our brief was to unify them into one coherent product that felt fast, precise, and worth returning to daily.

SCOPE

7 core surfaces

Home, Generate, Explore, Gradient, Library, Accessibility, History

OUR ROLE

Research through UI

Persona, sitemap, user flows, wireframes, and full visual design system

TARGET USER

Designers and developers

Fast-moving, keyboard-first, building for real client work

PLATFORM

Web application

Dark-mode-first, built for long daily sessions in the browser

THE BRIEF

Powerful tools, but no reason to remember which app they lived in

Colooors had real depth: an AI-assisted generator, a curated palette explorer, a gradient builder, a full color library, and a WCAG accessibility checker. The problem was not capability, it was cohesion. Each tool felt like its own product.

01 No obvious starting point

New visitors landed on a features page, not a tool, so the core value took too long to reach.

02 Generation needed to feel instant

The spacebar-to-generate mechanic was delightful once discovered, but nothing on the homepage surfaced it.

03 Accessibility was buried

WCAG contrast checking is a genuine differentiator, but it lived three clicks deep in the navigation.

04 Tools did not talk to each other

A palette generated in one tool could not be checked for accessibility without re-entering every hex code by hand.

Four phases, one connected toolkit

We treated every tool as one product with different modes, not eight separate destinations, so speed and consistency carried across the whole experience.

01

Discover

Usage analytics and designer interviews to see which tools were used together, and which were abandoned.

02

Define

Persona, sitemap, and user flows built around one continuous generate, check, export loop.

03

Design

Low-fidelity wireframes validated navigation and tool structure before any visual design began.

04

Deliver

A dark-mode-first visual system with one shared component library across every tool.

Designing around a designer who already has too many tabs open

JA

Jordan Ade

THE PRACTICAL DESIGNER

AGE

29

ROLE

Product Designer

TOOLS

Figma, Tailwind
CSS

DEVICE

Desktop, dark
mode

TEAM SIZE

5 to 10 people

BUDGET

Free tools first

GOALS

- Generate a palette that works across both light and dark UI
- Confirm WCAG compliance without manually checking every pair
- Export tokens straight into Tailwind config or CSS variables

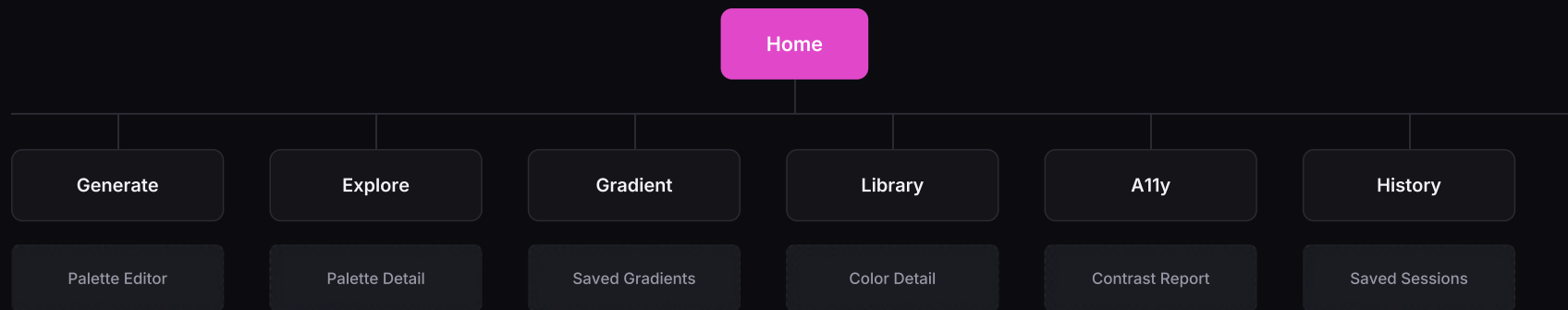
FRUSTRATIONS

- Switching between separate tools for generating and checking colors
- Re-typing hex codes that were already generated somewhere else
- Losing track of a good palette from a project two weeks ago

NEEDS FROM THE PRODUCT

- One flow that generates, checks, and exports without leaving the page
- A history that quietly saves every session automatically
- Keyboard shortcuts fast enough to match his own working speed

A sitemap built around one continuous workflow



SEARCH, CMD K

SAVE PALETTE

LIGHT, DARK TOGGLE

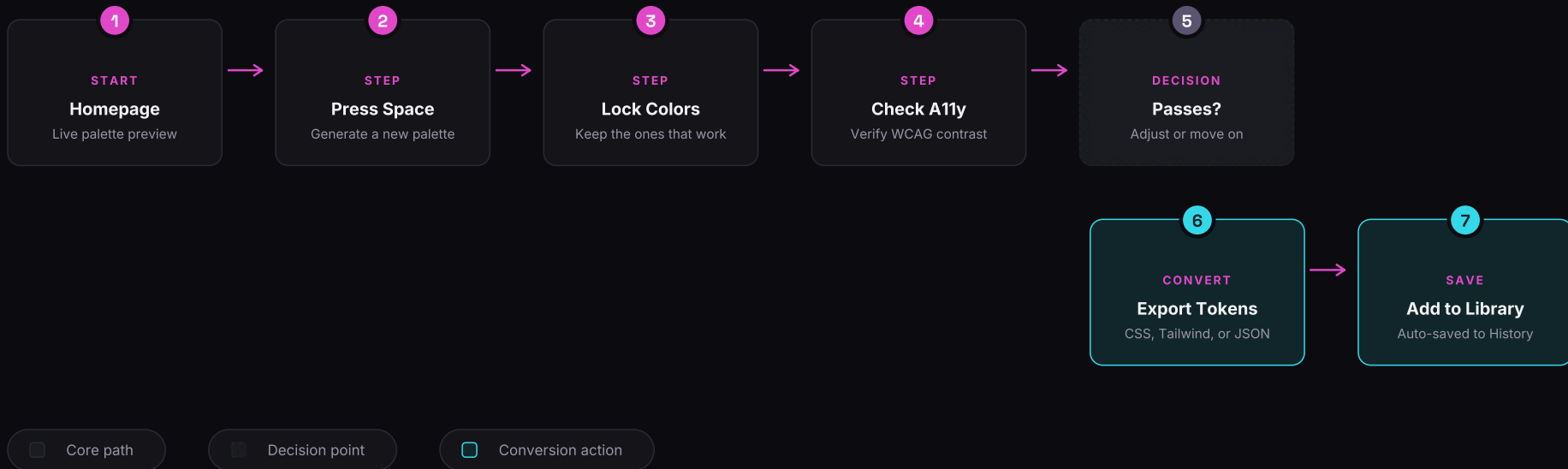
PERSISTENT UTILITY ACTIONS, AVAILABLE FROM EVERY TOOL

Following Jordan from first visit to a saved, accessible palette

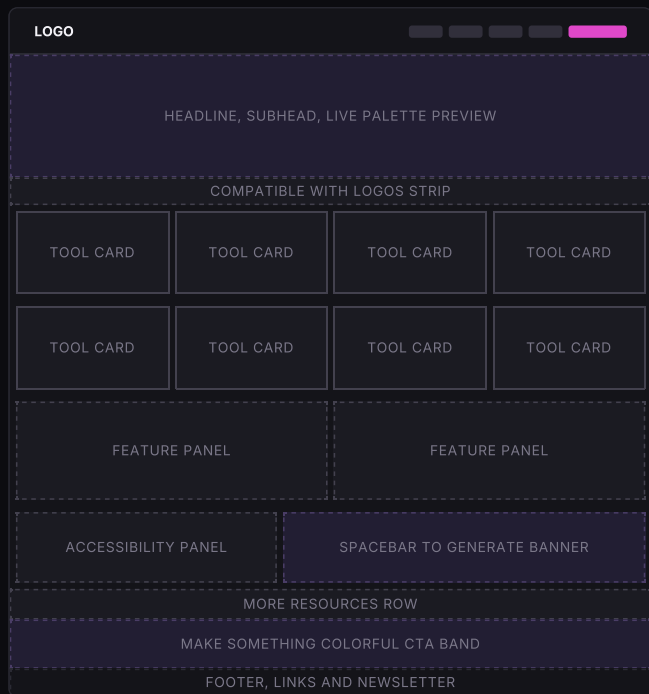
STAGE	GOAL	ACTIONS	TOUCHPOINTS	DESIGN RESPONSE
Awareness	Judge whether this tool is faster than his current one	Lands on homepage, sees a live palette preview	Hero, live color demo, tool grid	The generator is visible and interactive within the first screen
Discovery	Generate a palette that fits his project	Presses space to generate, locks favourite colors	Generate tool, keyboard shortcuts	Full-bleed color bars make generation feel immediate and tactile
Consideration	Confirm the palette is accessible	Opens the A11y checker on the same colors	Accessibility checker, live preview panel	Contrast results shown instantly against real UI text and buttons
Decision	Get the palette into his codebase	Exports as CSS variables or Tailwind config	Export panel, copy-to-clipboard actions	One-click export in the exact format his stack already uses
Retention	Come back without losing past work	Revisits History, browses Explore for new ideas	History log, curated palette gallery	Every session is saved automatically with no extra action required

The primary generate, check, export loop

Mapped before any visual design began, so a designer could go from a blank screen to an exported, accessible palette without ever leaving the page.



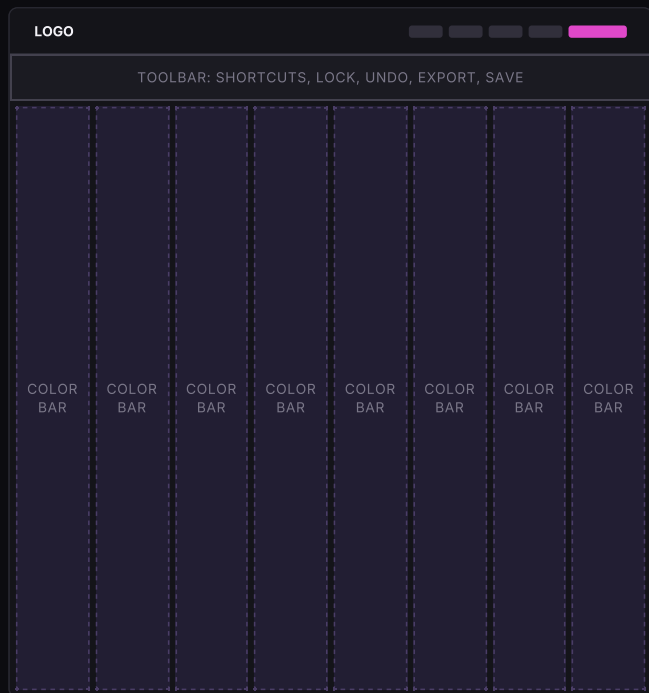
Homepage: structure before style



STRUCTURAL NOTES

- A live, interactive palette sits directly in the hero, so the core value is visible before any scrolling
- Eight tool cards are grouped in a tight grid rather than a long feature list, keeping the whole toolkit scannable at once
- The spacebar-to-generate mechanic gets its own dedicated banner, not just a mention in a feature card
- Accessibility is promoted to its own panel on the homepage instead of being buried in navigation

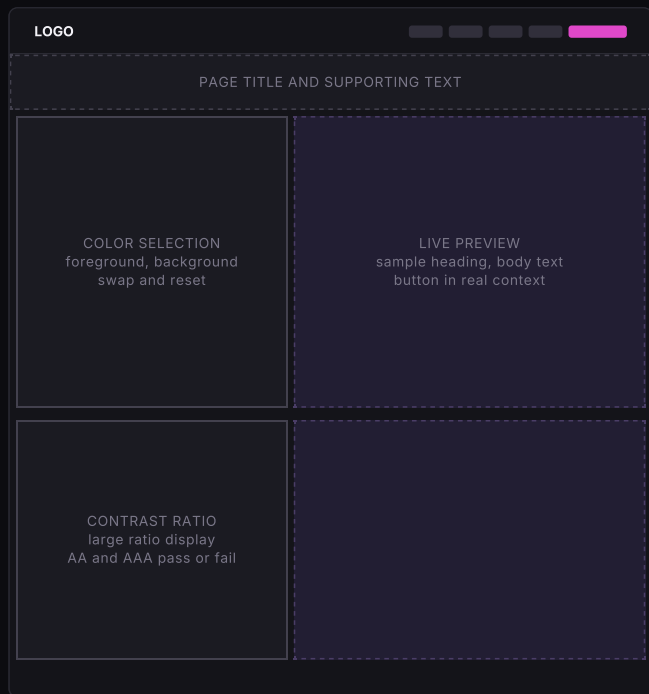
Generate: structure before style



STRUCTURAL NOTES

- Full-height color bars make the palette itself the entire interface, with almost no chrome competing for attention
- The toolbar stays slim and fixed at the top, keeping lock, export, and save always within reach
- Each bar shows its hex and name only on interaction, keeping the default view clean and visual
- The same eight-bar layout is reused for saved palettes in History and Explore, for instant familiarity

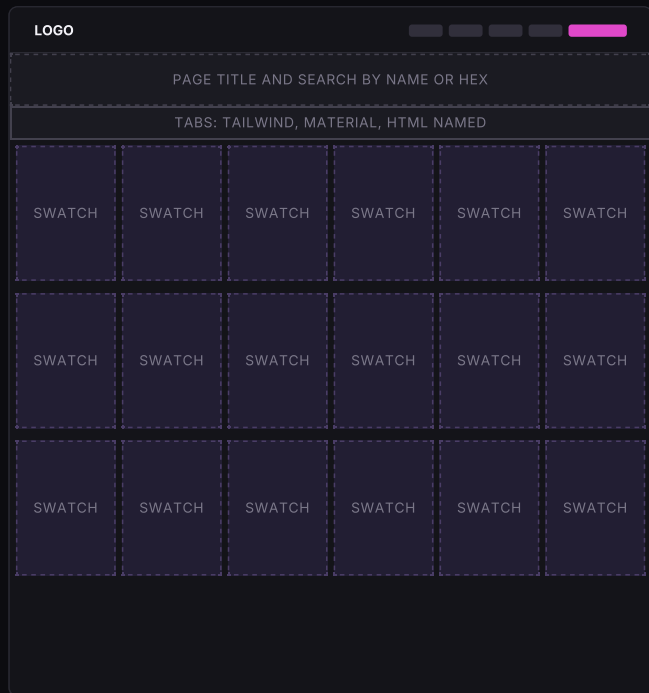
Accessibility checker: structure before style



STRUCTURAL NOTES

- Color selection and live preview sit side by side, so any change is reflected instantly without a submit step
- The contrast ratio is shown as one large, unmistakable number rather than a buried technical detail
- AA and AAA pass or fail states use consistent status colors matched to the rest of the product
- The same live preview pattern is reused wherever color decisions need instant, real-context feedback

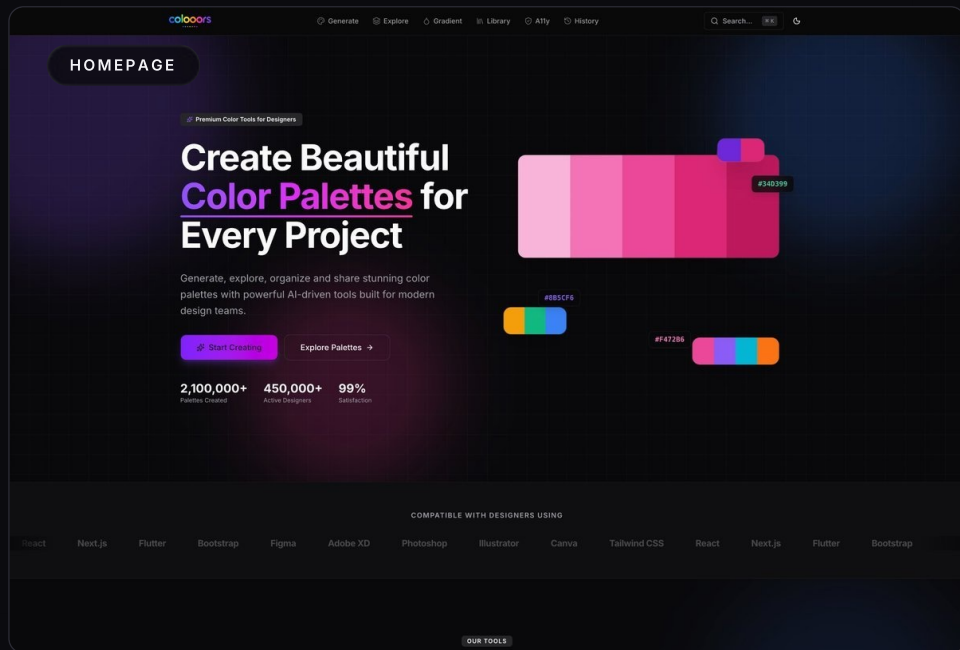
Color library: structure before style



STRUCTURAL NOTES

- Tabs split the library by standard (Tailwind, Material, HTML named) instead of mixing systems in one long list
- A dense six-column swatch grid favours fast visual scanning over reading names one at a time
- Search works by name or hex, so a developer arriving with a code value can find a match immediately
- Every swatch uses the same card pattern as Generate and Explore, keeping color always presented the same way

The homepage: the toolkit, not just a features page



A live palette in the hero, not a screenshot

Real, generated color bars sit next to the headline, so the product demonstrates itself before a visitor reads a single feature.

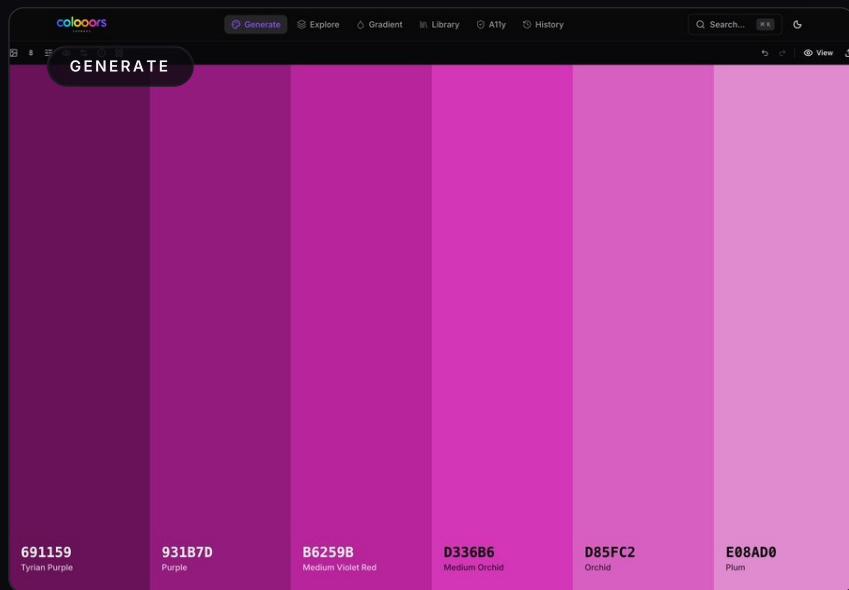
Eight tools, one tight grid

Every tool is presented as an equal, scannable card rather than a long marketing page, respecting that this audience wants to get to work.

Dark mode as identity, not a setting

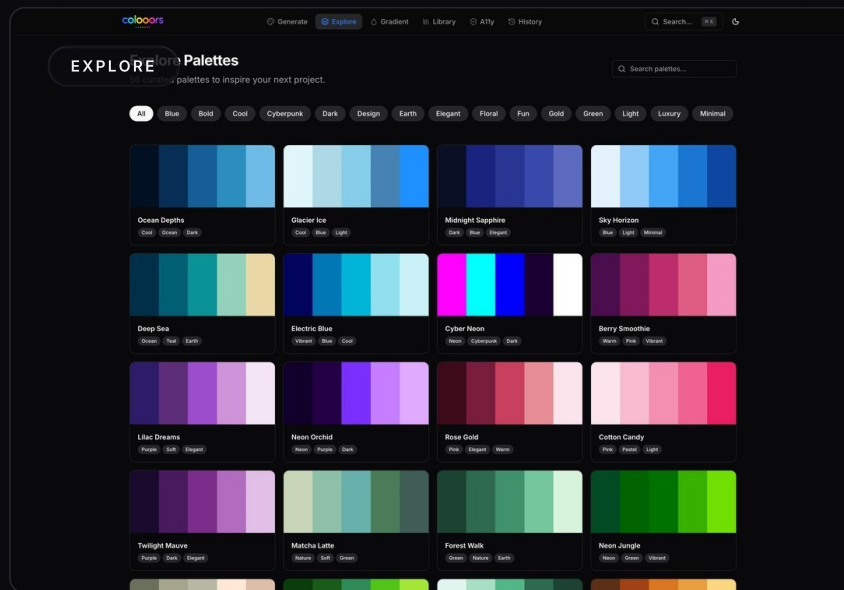
The interface stays dark throughout, matching the editors and code tools this audience already lives in all day.

Generation: the palette is the entire interface



Full-bleed color bars

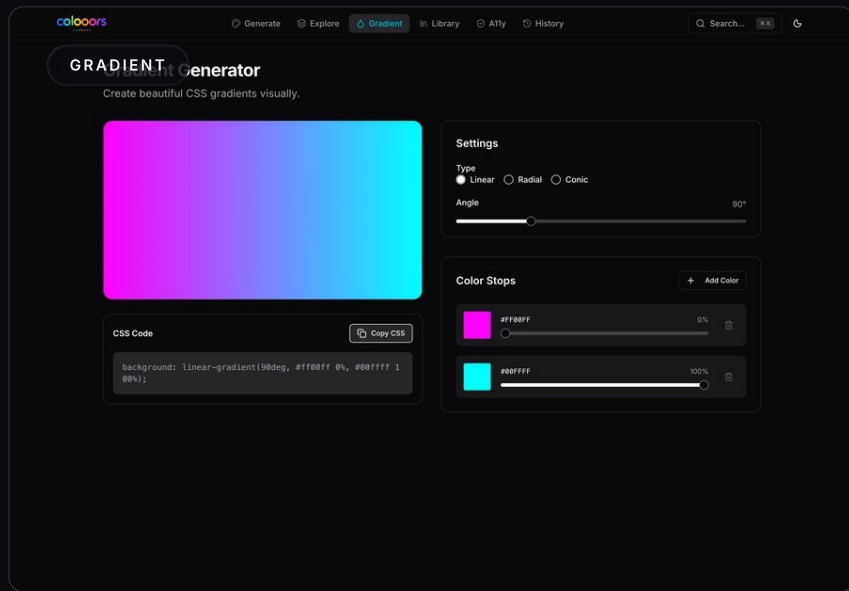
Chrome is stripped back to a single toolbar, so pressing space and watching eight new colors appear feels immediate and satisfying.



Explore: curated, not crowdsourced clutter

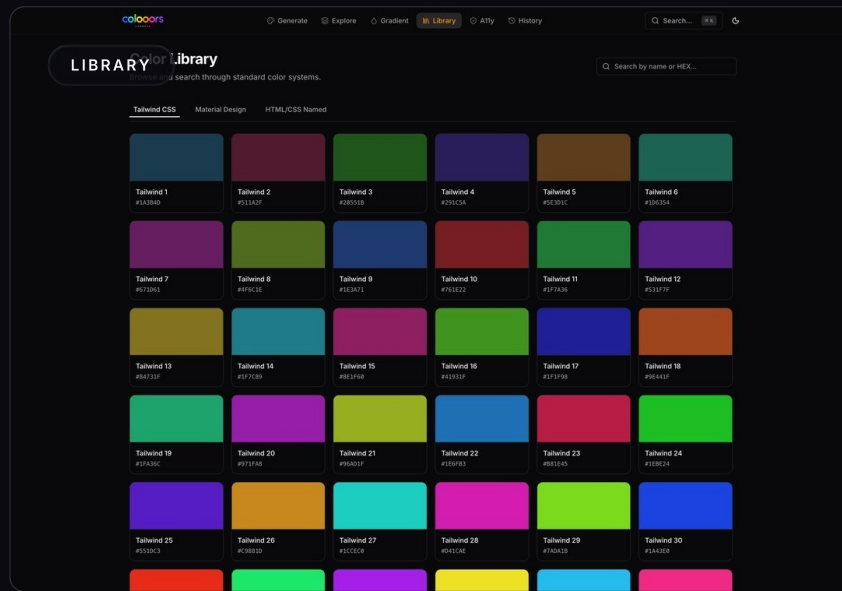
Tag filters (Cool, Bold, Luxury, Minimal) help designers browse by mood the same way they would brief a project.

Precision tools for the details that matter



Gradient generator with live CSS output

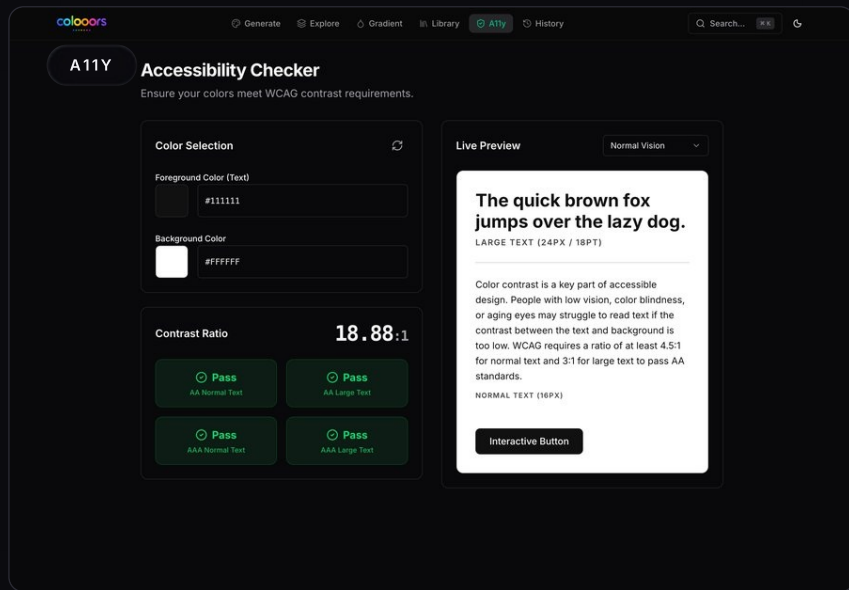
Every adjustment updates the canvas and the copy-ready CSS simultaneously, so there is never a translation step from design to code.



A complete reference library

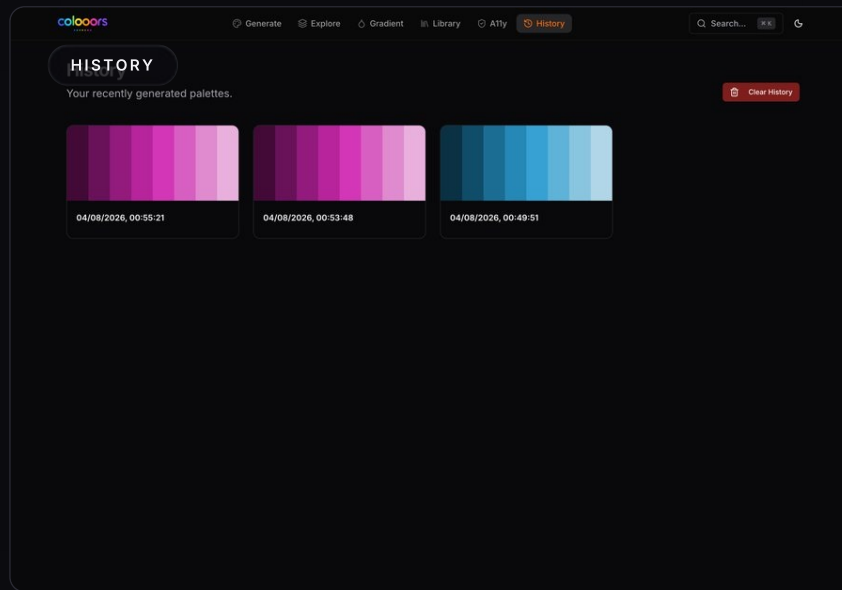
Tailwind, Material Design, and HTML named colors live in one searchable place instead of three separate browser tabs.

Trust and memory, built into the product



Accessibility made unmissable

A single large contrast ratio, tested against real UI text and a real button, turns a compliance chore into an instant answer.



History as a safety net

Every generated palette is saved automatically with a timestamp, so a good idea from last week is never truly lost.

The decisions behind the design



Dark mode as the default identity

A near-black canvas makes every generated color read at full saturation, and matches the editors this audience already works in.



Geometric type for a technical audience

A confident geometric sans for headlines signals precision and modernity, while a neutral grotesque keeps dense UI text easy to scan.



Accessibility promoted, not hidden

Contrast checking is surfaced on the homepage and reused as a live preview pattern, treating it as a feature rather than a checkbox.



A gradient accent used sparingly

The violet, magenta, and cyan gradient from the logo appears only at key moments, so it reads as a signature rather than noise.



One card system, everywhere

Palettes, gradients, and library swatches all share the same card anatomy, so switching tools never feels like switching products.



Keyboard shortcuts as a first-class path

Space to generate and Cmd K to search are designed as primary interactions, not hidden power-user tricks.

What the redesign was built to achieve

Objectives the design was built around: the levers a considered UX and visual system pull for a product like Colooors.

01

A faster path from landing to first palette

Surfacing live generation on the homepage removes the gap between arriving at the site and experiencing the core value.

02

One workflow instead of eight destinations

Shared navigation and a consistent card system let a designer move between tools without re-learning an interface each time.

03

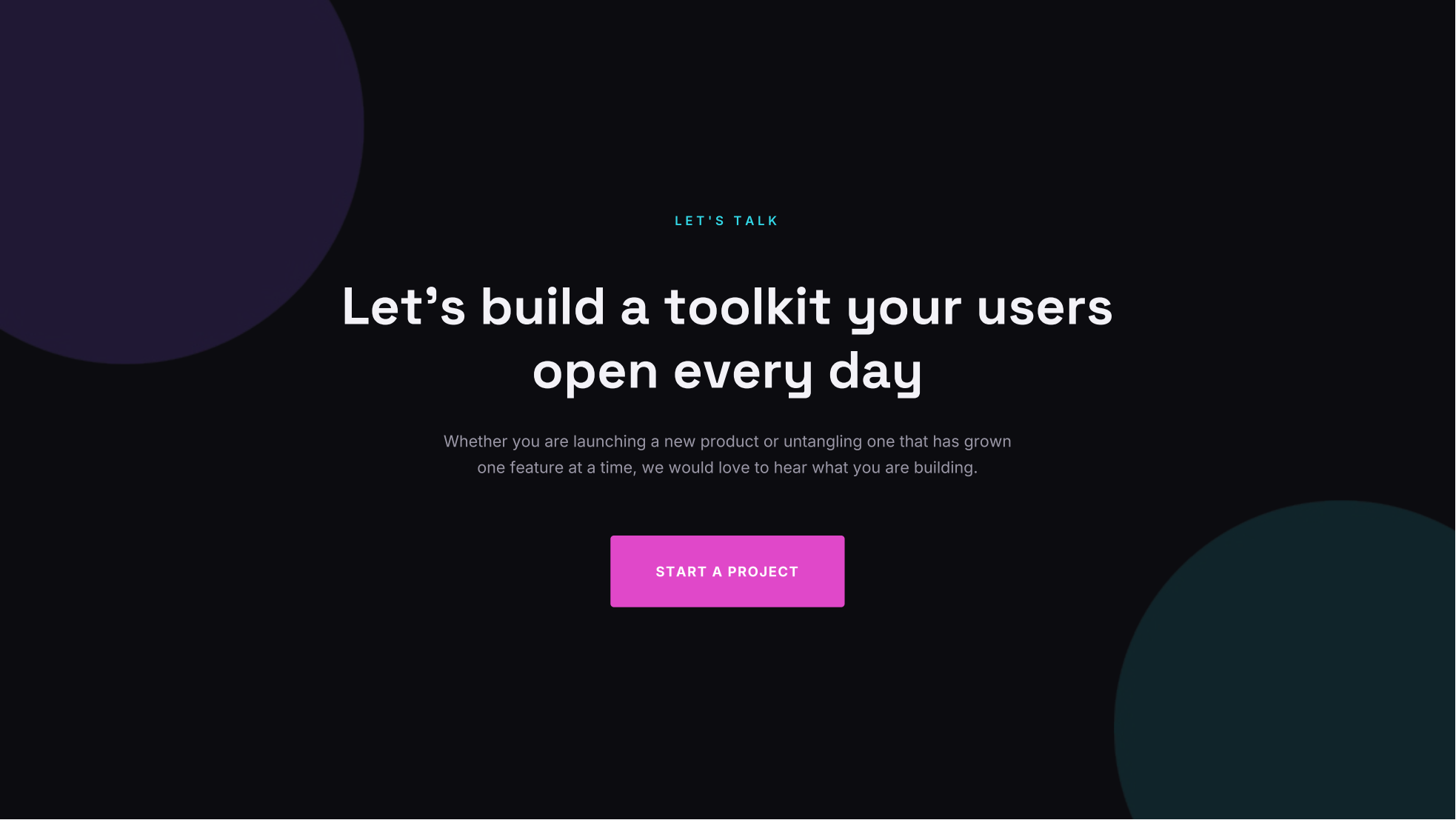
Accessibility as a selling point

Promoting the WCAG checker turns a compliance requirement into a visible reason to choose Colooors over a simpler generator.

04

A product built for daily, repeat use

Automatic history and fast keyboard-first interactions reward the kind of frequent, habitual use that free tools rarely earn.



LET'S TALK

Let's build a toolkit your users open every day

Whether you are launching a new product or untangling one that has grown one feature at a time, we would love to hear what you are building.

START A PROJECT