

[Open in app](#)



Generative AI

Member-only story

Here's How I Use Kombai To Build Frontend Fast

Why I think Kombai is the most useful tool for frontend devs right now



Jim Clyde Monge

[Follow](#)

12 min read · 3 days ago

113

2

+



Kombai

I've been doing front-end development for over a decade, and one thing I can say for sure is that building effective landing pages and dashboards used to take a lot of time. On average, it could take me anywhere from three days to a full week to complete a dashboard. It wasn't only about making things look good. Responsiveness, reusability of components, and overall maintainability also took a huge chunk of my time.

Fast forward to today, and AI coding agents have completely changed that.

Instead of manually building everything from scratch, I can simply describe the look and feel of a UI in plain words, and the AI writes the code that matches my vision. That's exactly how I built the landing pages and dashboards of my recent web apps like [Pixono](#), [Flux Labs AI](#), and [ReelPal](#).

There are already several AI-powered frontend platforms out there. Tools like Cursor, Lovable, and Bolt are all capable of generating front-end code, though they rely heavily on whichever LLM they're connected to.

In this article, I want to focus on a tool called [Kombai](#). It's an AI agent built specifically for frontend development tasks.

Let's get started.

What is Kombai?

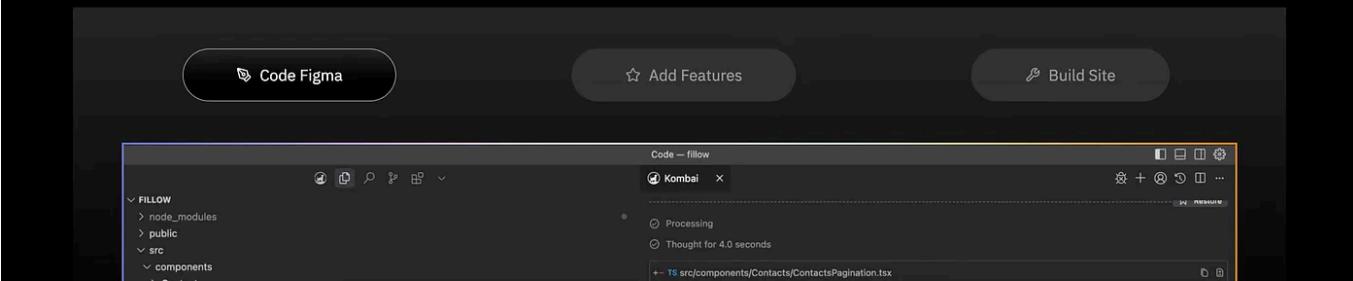
Kombai is an AI-powered extension for VS Code that helps you build beautiful and functional frontends using different inputs like Figma, text, or even images. It understands your codebase, follows best practices from 30+ frontend libraries, and produces clean, production-ready code.

The AI Agent Built for Frontend Development

Generic AIs struggle with complex frontend tasks. With specialized context-engine and tooling, Kombai delivers unmatched fidelity, code quality, and, dev velocity.

Install Kombai

Supports & 25+ Libraries



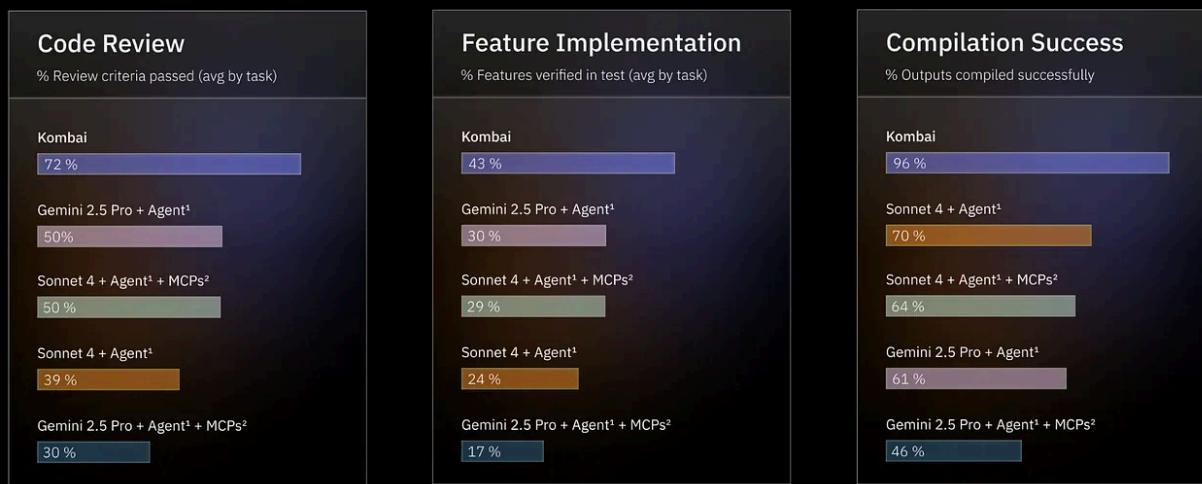
Kombai frontend generator. Image by [Jim Clyde Monge](#)

Unlike general-purpose coding agents, Kombai was tested on a custom benchmark of over 200 real-world tasks, covering both design-to-code and text-to-code conversions across different frameworks.

The evaluation measured three key factors:

1. Whether the code compiles without errors.
2. Whether it follows best practices such as reusability and separation of concerns.
3. Whether it delivers the expected design and functionality.

Benchmarks for Real-world Frontend Tasks



1. Agents used: Aider when without MCP, OpenAI agent + CodeMCP, OR Filesystems MCP when with MCP. All agents were allowed 3 attempts to fix errors.

2. MCPs used: Context7 for documentation, Framelink for Figma designs.

Kombai performance benchmarks. Image by [Jim Clyde Monge](#)

The results show that Kombai consistently performs better than general-purpose agents such as Claude Sonnet 4 or Gemini 2.5 Pro when it comes to frontend tasks.

The code compiles cleanly more often, passes structured code reviews at a higher rate, and produces outputs that match both the design and the expected functionality.

Kombai Installation

To install Kombai, go to its [official website](#) and hover over the Install Kombai button on the upper right corner of the page. You'll see options for which coding IDE you want to install it on.

Currently, it supports VS Code, Cursor, Windsurf, and Trae.

The AI Agent Built for Frontend Development

Generic AIs struggle with complex frontend tasks. With specialized context-engine and tooling, Kombai delivers unmatched fidelity, code quality, and, dev velocity.

Install Kombai    

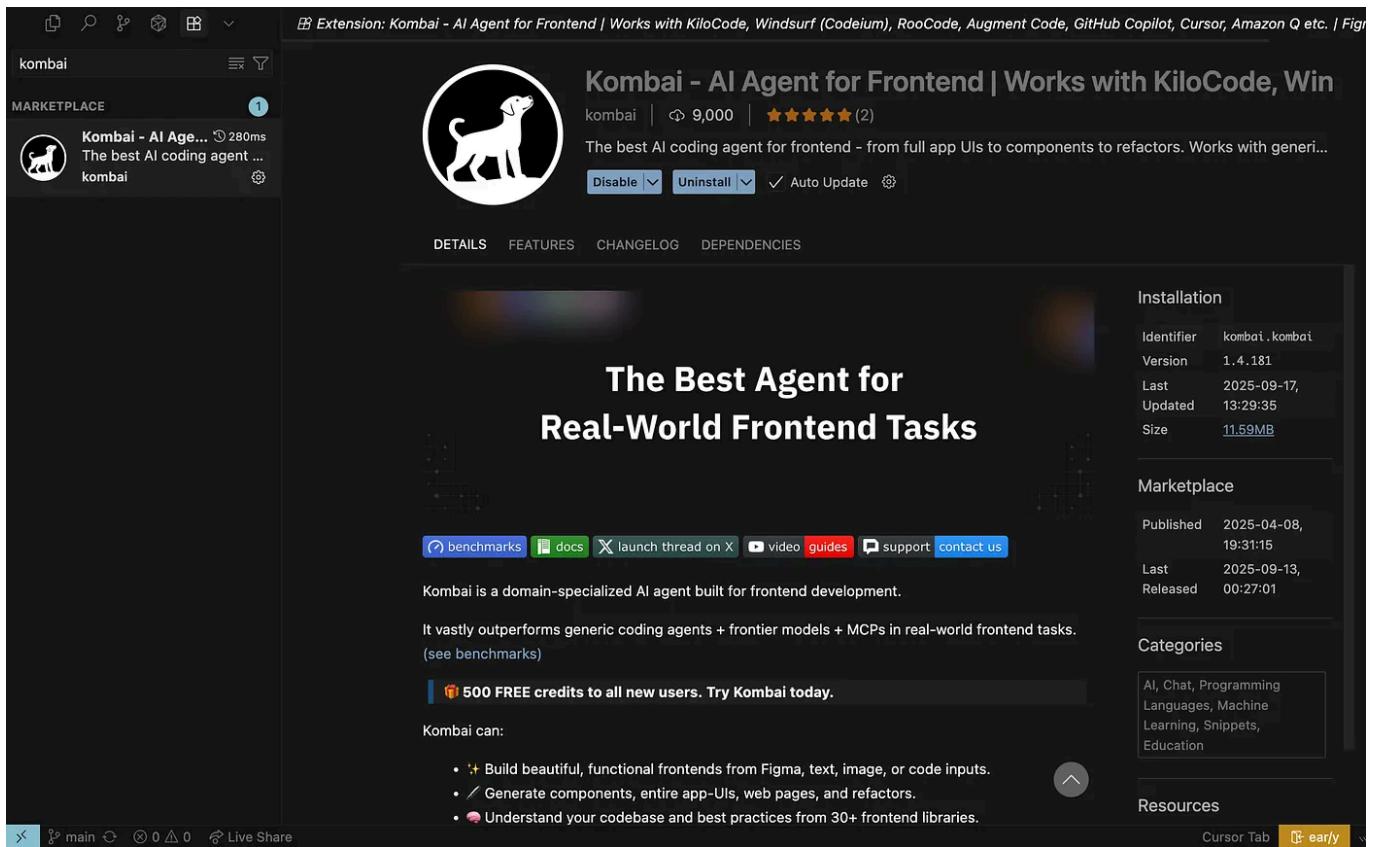
Supports



& 25+ Libraries

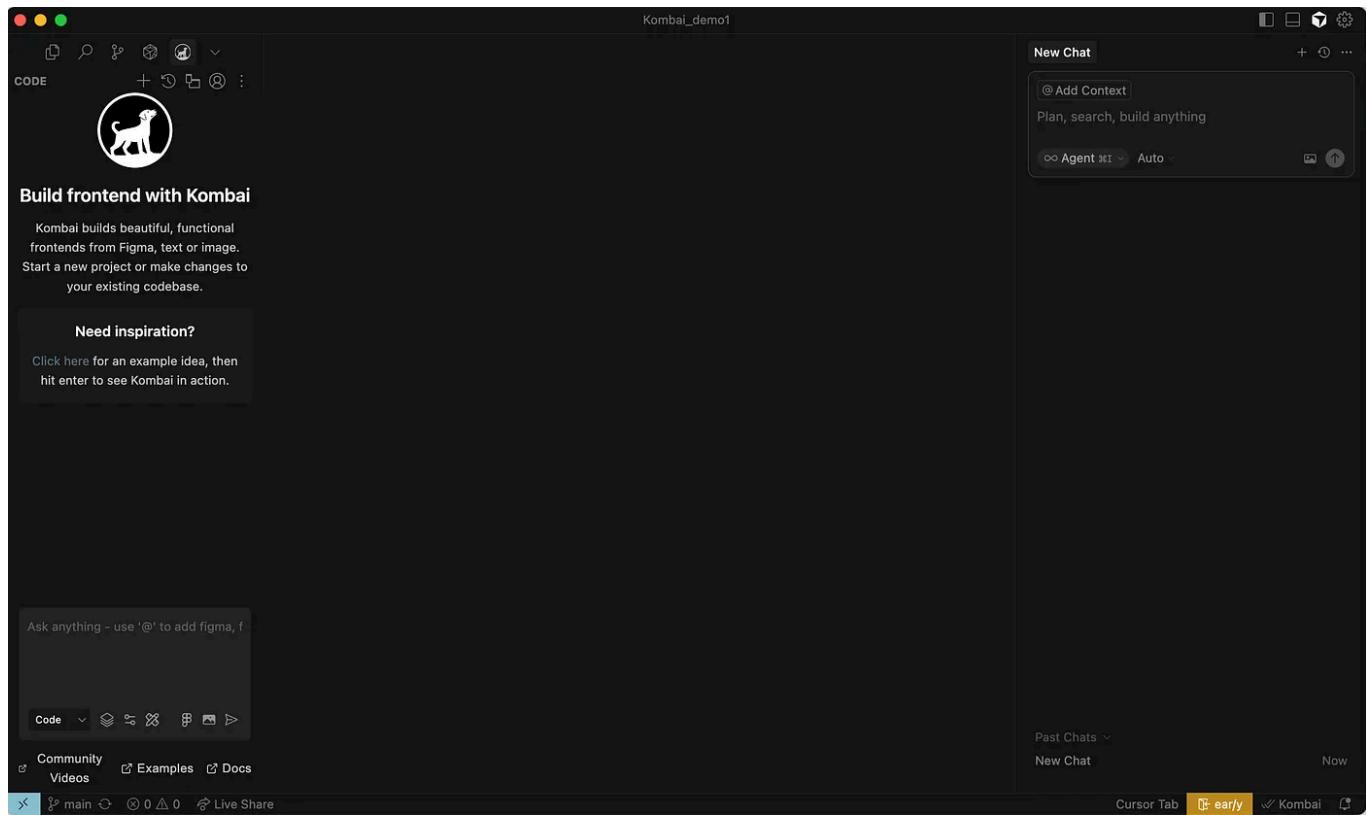
Kombai landing page. Image by [Jim Clyde Monge](#)

Here's an example screenshot of Kombai installation in the Cursor IDE.



Kombai installation dashboard. Image by [Jim Clyde Monge](#)

Feel free to explore the details and click on the Install button to download and integrate the plugin in Cursor. Once done, you should notice in the left panel that a new tab has been added for Kombai.



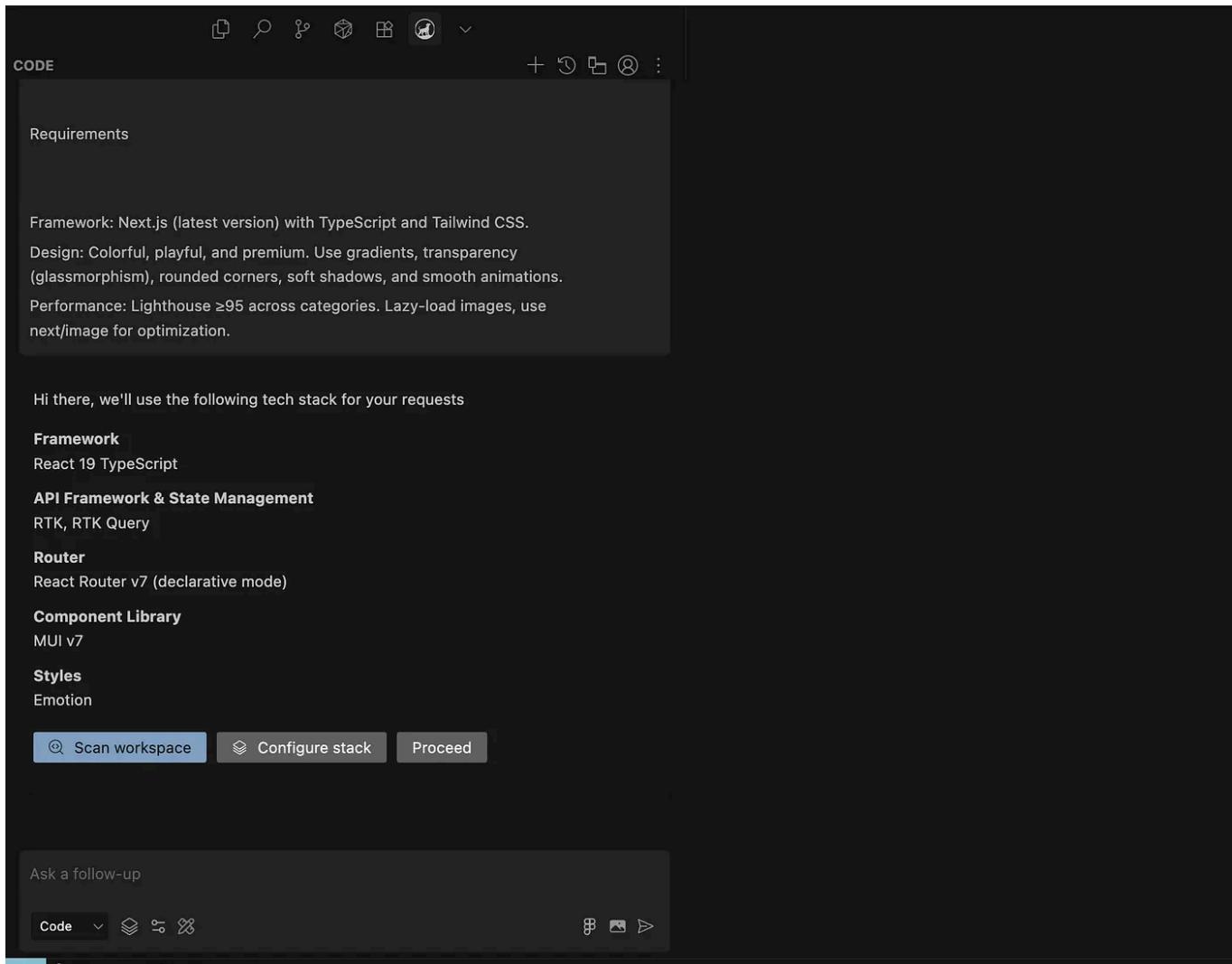
Kombai chat panel. Image by [Jim Clyde Monge](#)

That's it. You are now ready to build your first frontend module.

Building Frontends From Text Descriptions

To give you an example, I will try to create a landing page for my latest web application that creates children's storybooks using AI.

You can check the full prompt below. Paste it in the prompt field and then hit the submit button. Notice that Kombai does not immediately create files and write code. It chooses the appropriate tech stack and configuration that is subject to your approval.



Kombai chat example. Image by Jim Clyde Monge

Prompt:

Goal: Build a **Next.js** landing page **for** Noodloo AI, a children's storybook genera

Requirements

Framework: **Next.js** (latest version) **with** TypeScript **and** Tailwind CSS.

Design: Colorful, playful, **and** premium. Use gradients, transparency (glassmorphism), rounded corners, soft shadows, and smooth animations.

Performance: Lighthouse ≥ 95 across categories. Lazy-load images, use `next/image` for optimization.

Accessibility: WCAG 2.1 AA, semantic HTML, keyboard navigation, aria-* labels. R

Branding & Visuals

Mood: Whimsical, magical, kid-friendly.

Colors: Bright gradients (pink, purple, blue, green, yellow) **with** dark ink backg

Typography: Rounded playful font **for** headings (e.g., Fredoka) + clean sans-serif

Decor: Floating blobs, soft particles, star/planet SVGs, morphing shapes.

Animations & Interactions

Smooth entrance fades/slides (350–450ms).

Hover lift/glow on buttons.

Floating particles in hero.

Scroll-triggered card reveals.

Sections to Include

Header: Logo + nav (Features, How It Works, Gallery, Pricing, FAQ, Order Print).

Hero: Headline “Create Magical Stories for Kids” + subheadline. CTA buttons: Sta

Features Grid:

AI-Powered Stories

Beautiful Illustrations

Custom Characters

Professional PDFs

Safe & Private

Share & Collaborate

How It Works: 3 steps (Imagine → Create → Make It Yours) with icons/mini illustrations.

Gallery / Art Styles: Horizontal scroll of sample illustration styles.

Custom Characters: Teaser card for character builder.

Download & Print: Emphasize PDF + printed copies. CTA: Order a Printed Book.

Reviews/Testimonials: Rotating quotes from parents.

Pricing Preview: Free / Pro / Family cards.

FAQ: Accessible accordion.

CTA Band: “Ready to Create Magic?” with Start Creating button.

Footer: Links, social icons, newsletter signup.

Copy to Use (verbatim)

Create Magical Stories for Kids

Turn your child’s imagination into unique, beautifully illustrated storybooks that Everything You Need

Our platform provides all the tools to create, customize, and share amazing stories.

AI-Powered Stories

Generate unique, engaging stories tailored to your child’s imagination

Beautiful Illustrations

Every story comes with stunning, custom illustrations in multiple art styles

Custom Characters

Create and save characters that can appear in multiple stories

Professional PDFs

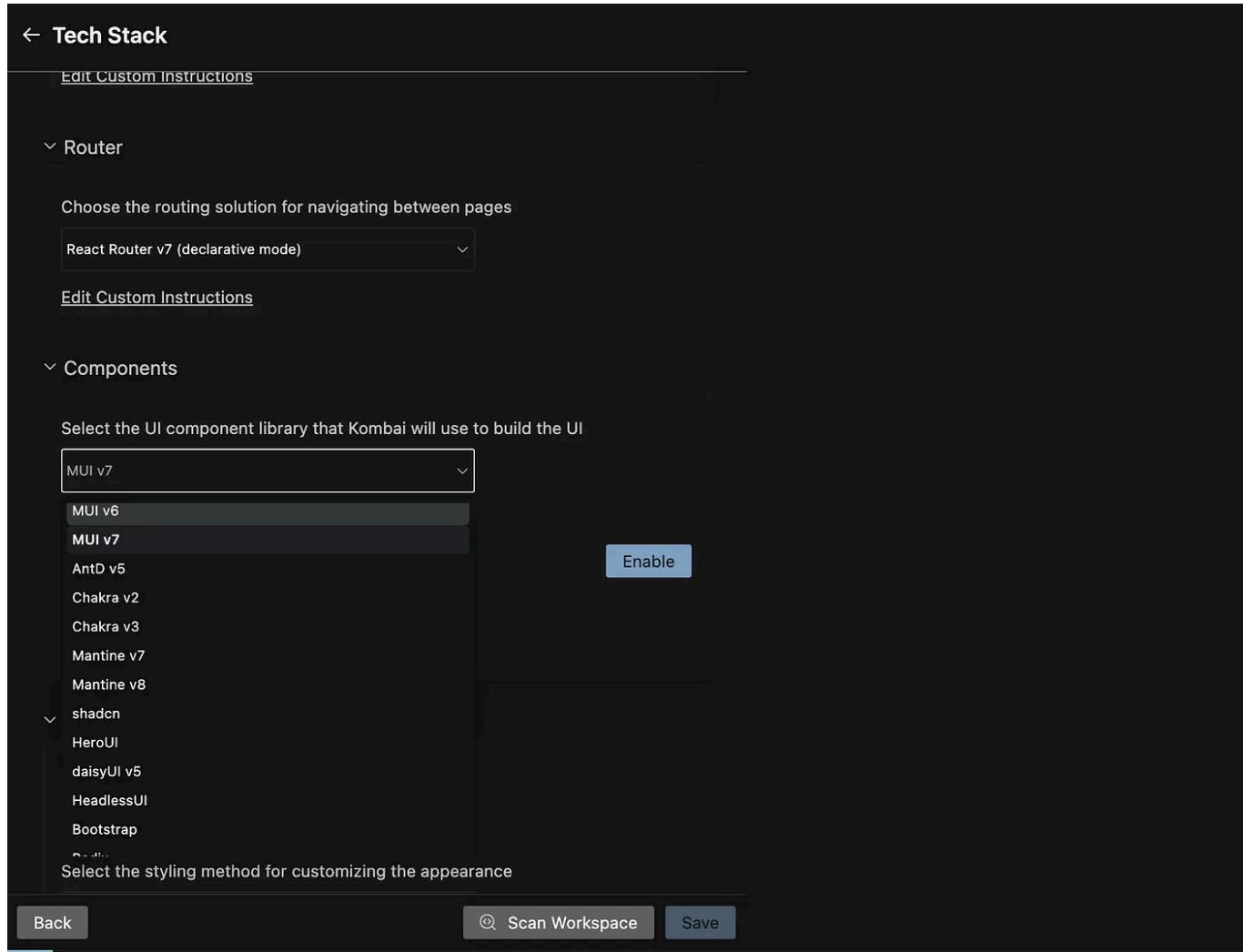
Download your stories as high-quality PDFs perfect for printing

Ready to Create Magic?

Join thousands of families creating unforgettable stories. Start your magical jo

Users can also order printed copies of their storybooks.

You can adjust the tech stack or change the UI configuration as you like. In this example, I want the routing solution to be React Router v7 and use MUI v7 for the UI components.



Kombai tech stack selection. Image by [Jim Clyde Monge](#)

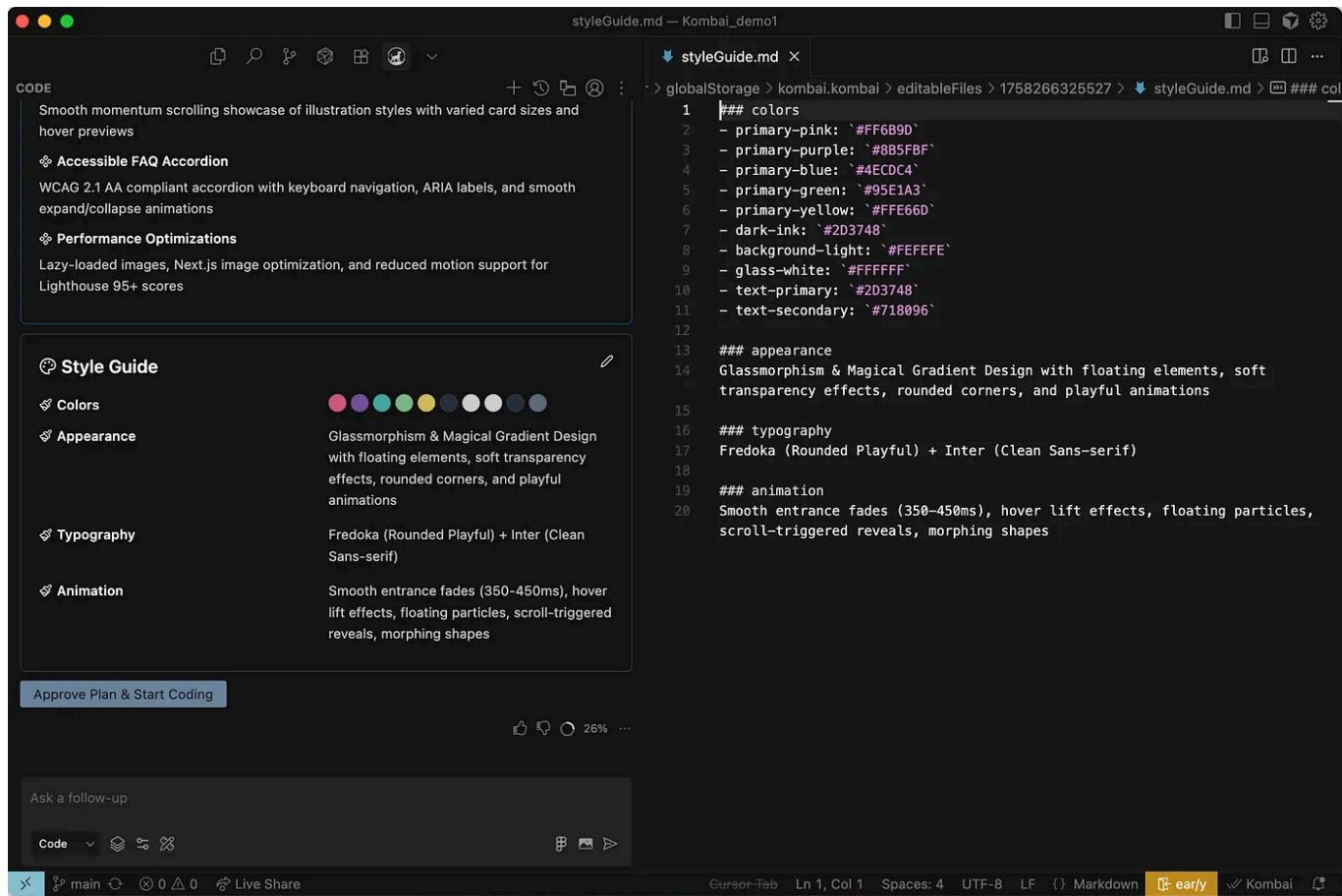
To customize things like the Feature list, FAQ, and other sections of the landing page, click on the little pencil button in the upper section and edit the content on the right side.

The screenshot shows the Kombai markdown editor interface. On the left, there's a 'CODE' pane containing sections like 'Ready to Create Magic?' and 'Footer'. On the right, a larger pane displays the raw markdown code for a file named 'features.md'. A red arrow points from the 'CODE' pane towards the right-hand code editor. The code editor has a red border around its content area. The code itself is a list of numbered items, each starting with a pound sign (#). The items include:

```
1  ### Glassmorphism Design System
2  Transparent cards with blur effects, soft shadows, and gradient borders
   throughout the interface for a premium magical feel
3
4  ### Floating Particle Animations
5  Interactive floating particles in hero section with smooth movement and
   hover interactions to create magical atmosphere
6
7  ### Scroll-Triggered Reveals
8  Cards and sections animate into view as user scrolls with staggered timing
   and smooth transitions
9
10  ### Responsive Mobile Menu
11  Hamburger menu with glassmorphism overlay and smooth slide animations for
    mobile navigation
12
13  ### Interactive Hover Effects
14  Buttons and cards lift with glow effects on hover, character previews
    morph, and pricing cards highlight dynamically
15
16  ### Horizontal Gallery Scroll
17  Smooth momentum scrolling showcase of illustration styles with varied card
    sizes and hover previews
18
19  ### Accessible FAQ Accordion
20  WCAG 2.1 AA compliant accordion with keyboard navigation, ARIA labels, and
    smooth expand/collapse animations
21
22  ### Performance Optimizations
23  Lazy-loaded images, Next.js image optimization, and reduced motion support
    for Lighthouse 95+ scores
```

Kombai markdown editing. Image by [Jim Clyde Monge](#)

The same is true with the Style Guide. You can control things like the color, typography, animation, etc.



Kombai markdown editing. Image by [Jim Clyde Monge](#)

After you edit the plan according to your preferences, click on the “Approve Plan & Start Coding” button.

This will trigger Kombai to start creating the files and writing the necessary code to build the frontend. Note that the agent also performs code build and fixes any build issues automatically.

Once the code generation is done, you can explore the files and inspect the code on the left panel.

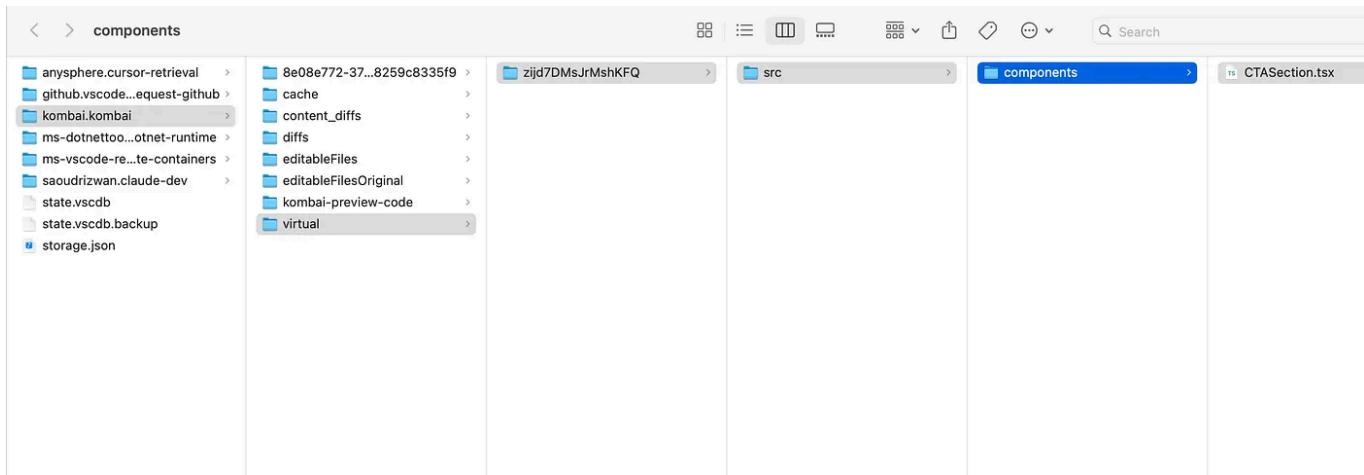
The screenshot shows the Kombai interface. On the left, a sidebar displays a tree view of files under 'src' and 'components'. A red arrow points to the 'Code generated' section in the sidebar. The main area shows a code editor with the file 'CTASection.tsx' open, containing TypeScript code for a component. A message at the top of the editor says: 'Fixed the TypeScript errors by refactoring GradientButton to use a proper React component with separate styled component, avoiding conflicts with MUI's Button variant prop.' Below the editor is a message: 'The generated code is in a sandbox. You can integrate the code and/or save them in your workspace. You can also run the code in the sandbox. See docs' with a 'Run in Sandbox' button. At the bottom, there are tabs for 'Code' and other options.

```
balStorage > kombai.kombai > virtual > zijd7DMsJrMshKFQ > src > components > CTATextSection.tsx
1 import React from 'react';
2 import { Box, Typography, Container, Stack } from '@mui/material';
3 import { styled } from '@mui/material/styles';
4 import GradientButton from './GradientButton';
5 import FloatingParticles from './FloatingParticles';
6
7 const CTAContainer = styled(Box)(({ theme }) => ({
8   position: 'relative',
9   padding: theme.spacing(12, 0),
10  background: 'linear-gradient(135deg, #8B5FBF 0%, #FF6B9D 25%, #FFE66D 50%, #8B5FBF 75%, #FF6B9D 100%)',
11  overflow: 'hidden',
12  '&:before': {
13    content: '',
14    position: 'absolute',
15    top: 0,
16    left: 0,
17    right: 0,
18    bottom: 0,
19    background: 'rgba(255, 255, 255, 0.1)',
20    backdropFilter: 'blur(1px)'
21  }
22 }));
23
24 const ContentBox = styled(Box)({
25   position: 'relative',
26   zIndex: 2,
27   textAlign: 'center',
28 });
29
30 const CTATitle = styled(Typography)(({ theme }) => ({
31   fontFamily: '"Fredoka", sans-serif',
32   fontWeight: 600,
33   fontSize: 'clamp(2.5rem, 6vw, 4rem)',
34   lineHeight: 1.2,
35   color: '#FFFFFF',
36   textShadow: '0 4px 20px rgba(0, 0, 0, 0.3)',
37   marginBottom: theme.spacing(3),
38 }));
39
```

Kombai code generation example. Image by [Jim Clyde Monge](#)

By default, the generated code is in a sandbox. That is why when you try to right-click and open the file in your local disk, you'll notice that they are not saved in your project folder. Instead, these files are saved in a virtual environment.

That's why when you view the generated files in your local disk, you'll notice that they are saved in a “virtual” folder.



Kombai code generation example. Image by [Jim Clyde Monge](#)

To preview the project, open the localhost link provided by Kombai. In my case, the server is running in <http://localhost:6321>

Fixed the TypeScript errors by refactoring GradientButton to use a proper React component with separate styled component, avoiding conflicts with MUI's Button variant prop.

The generated code is in a sandbox. You can integrate the code and/or save them in your workspace. You can also run the code in the sandbox. See docs

Run in Sandbox

✓ Server running at <http://localhost:6321/>

Preview doesn't match? Click to fix.

Code generated

View & Save

Ask a follow-up

Code

Preview icon

Save icon

Share icon

A red arrow points from the text "Server running at http://localhost:6321/" to the URL itself.

Kombai running example. Image by [Jim Clyde Monge](#)

This is what the landing page looks like:

Create Magical Stories for Kids

Turn your child's imagination into unique, beautifully illustrated storybooks that you can also order in print.

[Start Your Story](#) [See How It Works](#)

Everything You Need

Our platform provides all the tools to create, customize, and share amazing stories

- AI-Powered Stories**: Generate unique, engaging stories tailored to your child's imagination
- Beautiful Illustrations**: Every story comes with stunning, custom illustrations in multiple art styles
- Custom Characters**: Create and save characters that can appear in multiple stories
- Professional PDFs**: Download your stories as high-quality PDFs perfect for printing
- Safe & Private**: Your family's stories are kept secure with enterprise-grade privacy protection
- Share & Collaborate**: Share stories with family members and collaborate on creative adventures

Art Styles Gallery

Explore the beautiful illustration styles available for your stories

- Fantasy Adventure**: Magical creatures and fairy tale worlds
- Pirate Adventures**: High seas adventures with treasure maps
- Animal Friends**: Cute woodland creatures and forest adventures
- Space Exploration**: Cosmic adventures with rockets and planets
- Whimsical Tales**: Colorful characters in magical worlds

Ready to Create Magic?

Join thousands of families creating unforgettable stories. Start your magical journey today!

[Start Creating](#)

Noodloo AI Features How It Works Gallery Pricing FAQ Order Print

Noodloo AI Features How It Works Gallery Pricing FAQ Order Print

Noodloo AI Features How It Works Gallery Pricing FAQ Order Print

Noodloo AI Features How It Works Gallery Pricing FAQ Order Print

Awesome. I love the color choices, the big fonts, the layout, and the addition of placeholder images. The animation and responsiveness of the landing page are also on point.

At this point, I can make further changes to the source code whilst in the virtual environment. Just describe the customization changes you want to the coding agent. For example, I wanted the color theme to be darker.

Create Magical Stories for Kids

Turn your child's imagination into unique, beautifully illustrated storybooks that you can also order in print.

[Start Your Story](#)[See How It Works](#)

Everything You Need

Our platform provides all the tools to create, customize, and share amazing stories

AI-Powered Stories

Generate unique, engaging stories tailored to your child's imagination

Beautiful Illustrations

Every story comes with stunning, custom illustrations in multiple art styles

Custom Characters

Create and save characters that can appear in multiple stories

Professional PDFs

Download your stories as high-quality PDFs perfect for printing

Safe & Private

Our platform provides all the tools to create, customize, and share amazing stories

Share & Collaborate

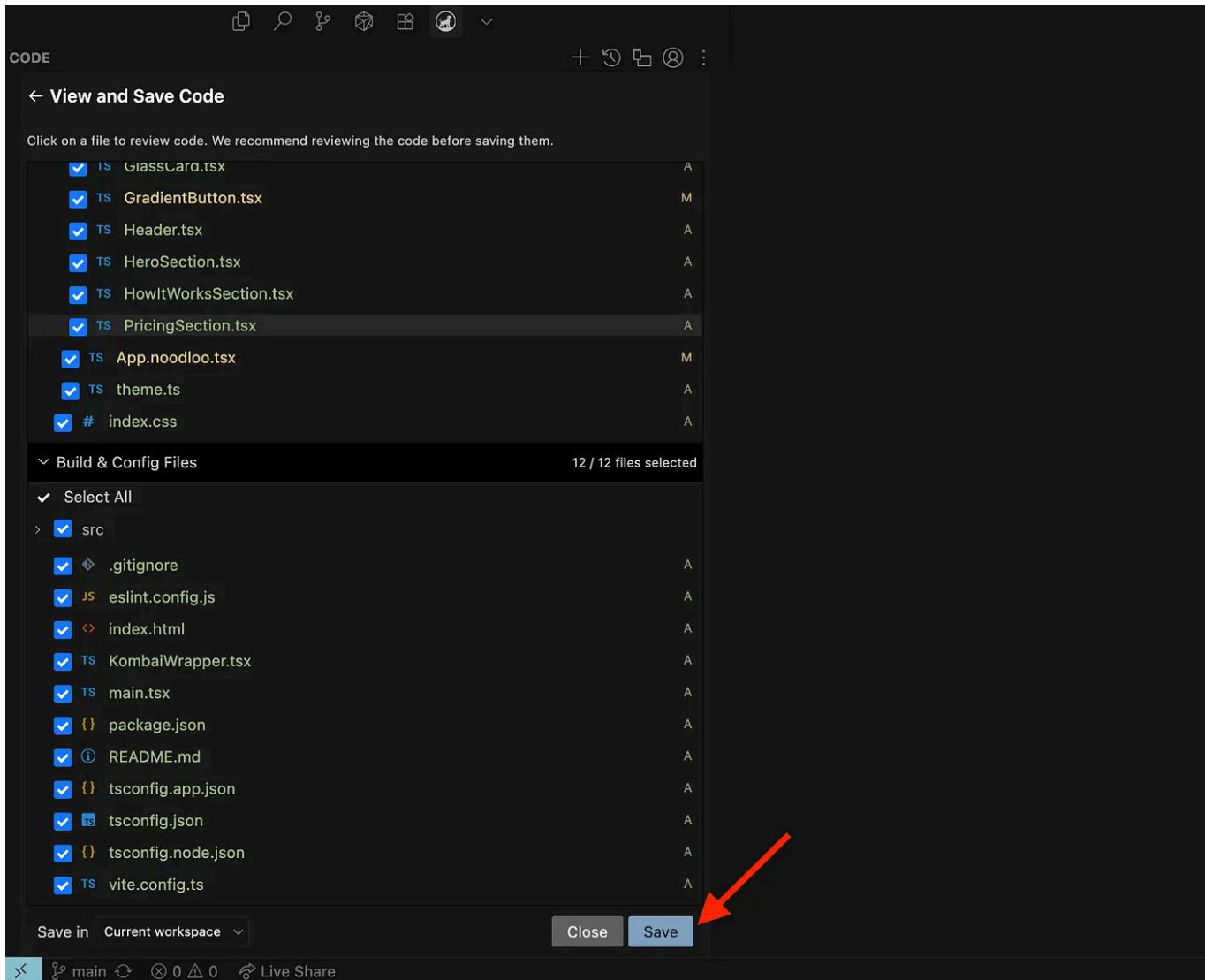
Our platform provides all the tools to create, customize, and share amazing stories

© 2025 Noodloo AI

Kombai generated code example. Image by [Jim Clyde Monge](#)

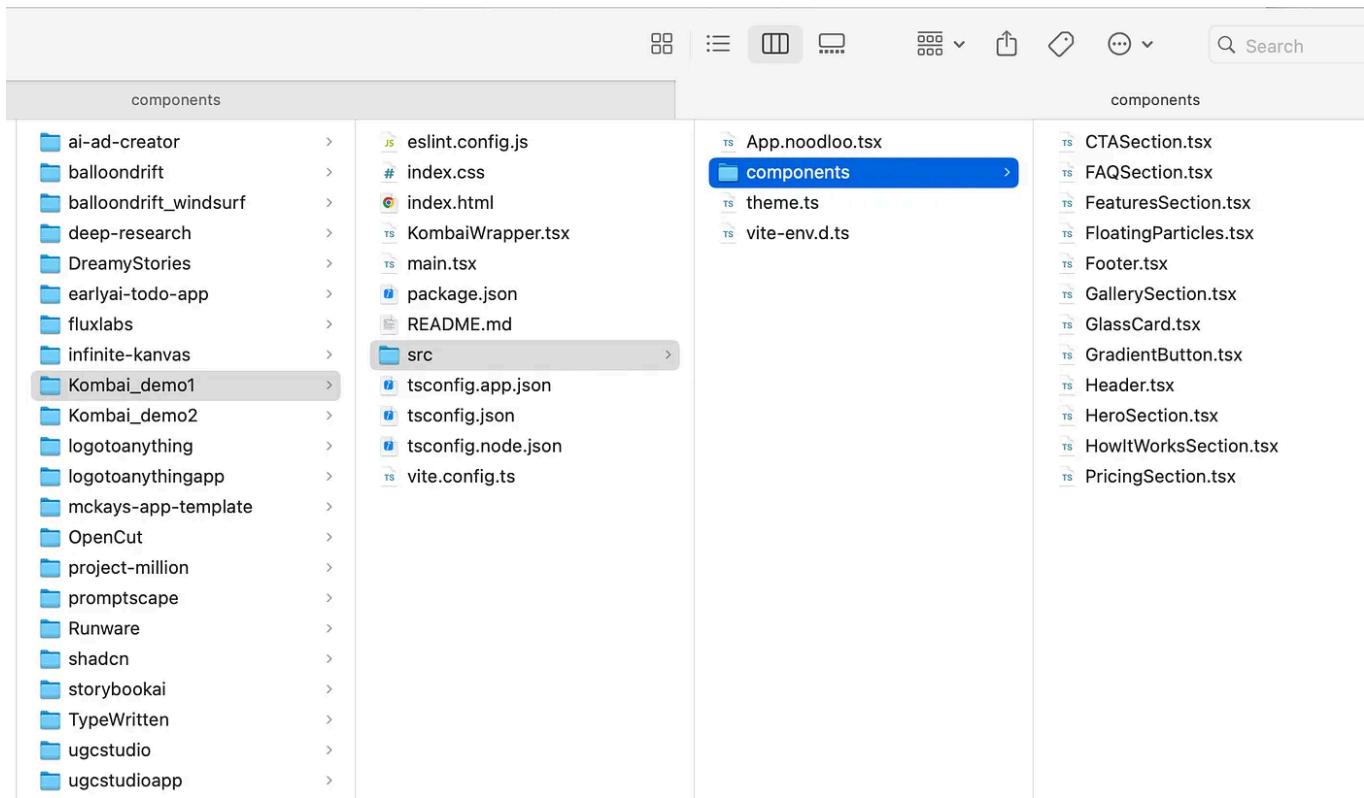
Prompt: Change the color theme to a darker tone and make the front and components more compact.

Once you're happy with the design and code, view the file list and click on the Save button.



Kombai generated code example. Image by [Jim Clyde Monge](#)

This will save the files to your actual working directory.



Kombai generated code example. Image by [Jim Clyde Monge](#)

That's it. You now have a working frontend module that's ready to get integrated with a backend.

Three things that I immediately realized after using Kombai:

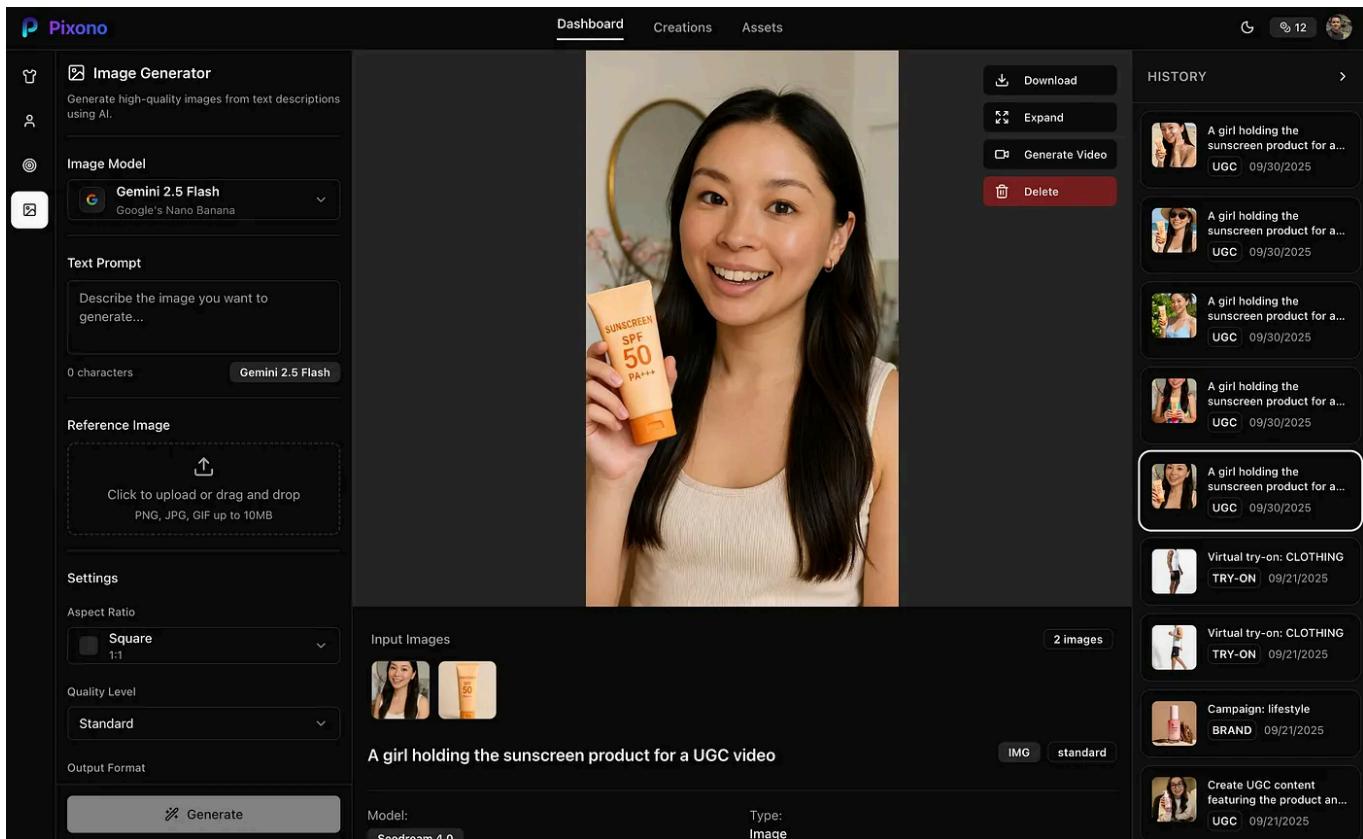
- 1. The ability to customize before generation.** The first thing I noticed is the ability to customize the tech stack and review the plan before the actual code is generated. This feature is very powerful because most vibe coding platforms skip this step, which often leads to wasted tokens and miscommunication.
- 2. The first build usually compiles perfectly.** With other tools, I usually end up stuck fixing errors caused by the AI, but with Kombai, the initial output almost always compiles successfully.
- 3. The third is the UI quality.** Apps generated by tools like Bolt or Lovable tend to look generic and too similar. With Kombai, the layout and design

feel much more in line with the text descriptions, and the results are easier to control.

In the next section, I will show you how to copy existing websites by taking a screenshot and letting Kombai transform it into a working code.

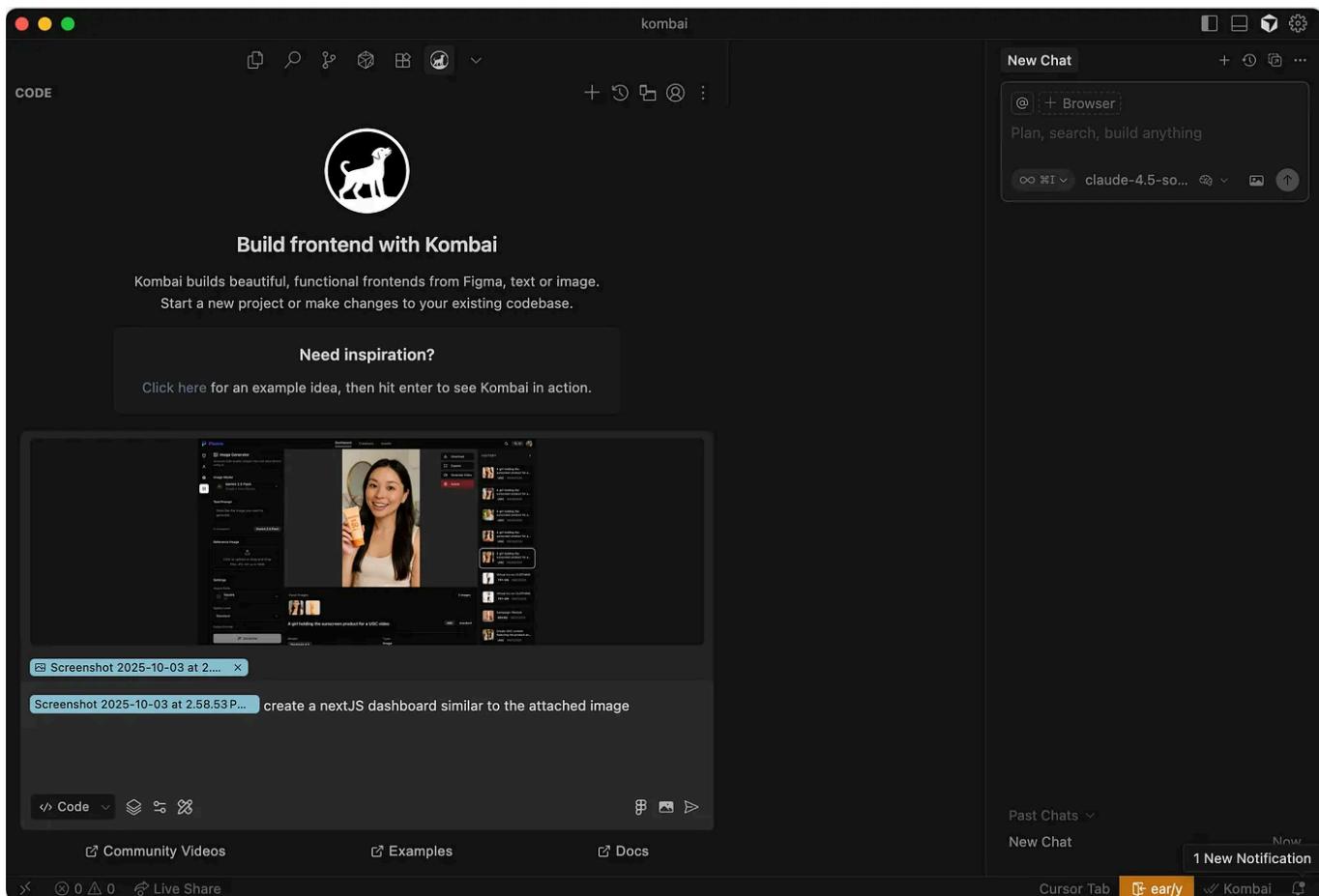
Creating Dashboards From a Screenshot

In this example, I wanted to copy an image generator dashboard from [Pixono](#). I took a screenshot and pasted the image into Kombai's prompt field.



Pixono dashboard screenshot. Image by [Jim Clyde Monge](#)

Prompt: Create a NextJS dashboard similar to the attached image



Plxono dashboard screenshot attached in Kombai. Image by [Jim Clyde Monge](#)

Again, Kombai will try to make a tech stack and execution plan for me.

Here's a snippet related to the layout and content:

Sections

Main Layout

The `main` dashboard layout with sidebar, `canvas` area, and history panel

Left Sidebar

`Left` navigation sidebar containing logo, navigation icons, image generator contr

Top Navigation

`Top` navigation bar with Dashboard/Creations/Assets tabs and user profile `section`

Canvas Area

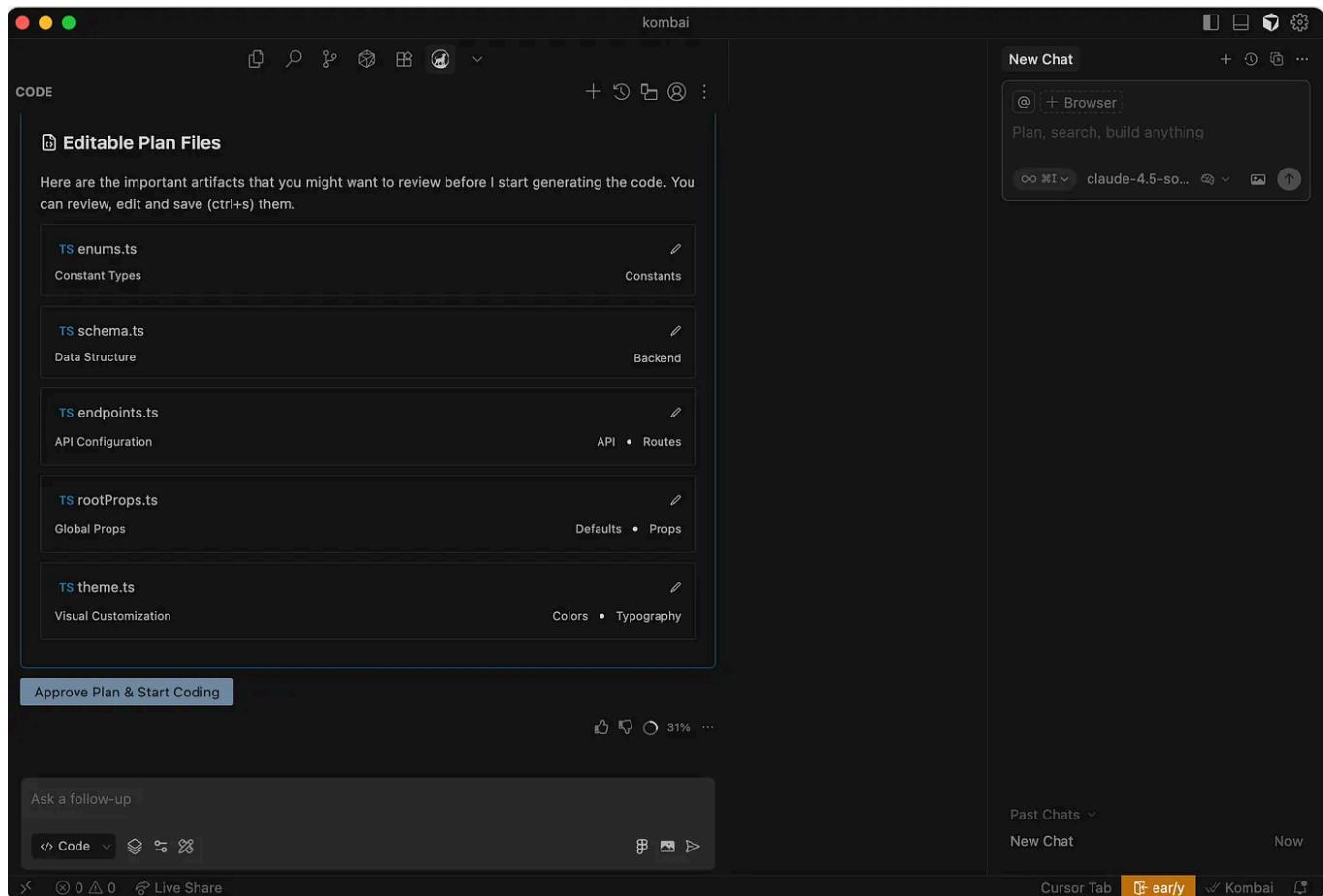
Central `canvas` area displaying the generated image with action buttons (Download, Image Details)

`Bottom section` showing `input` images, description, model info, and type

History Panel

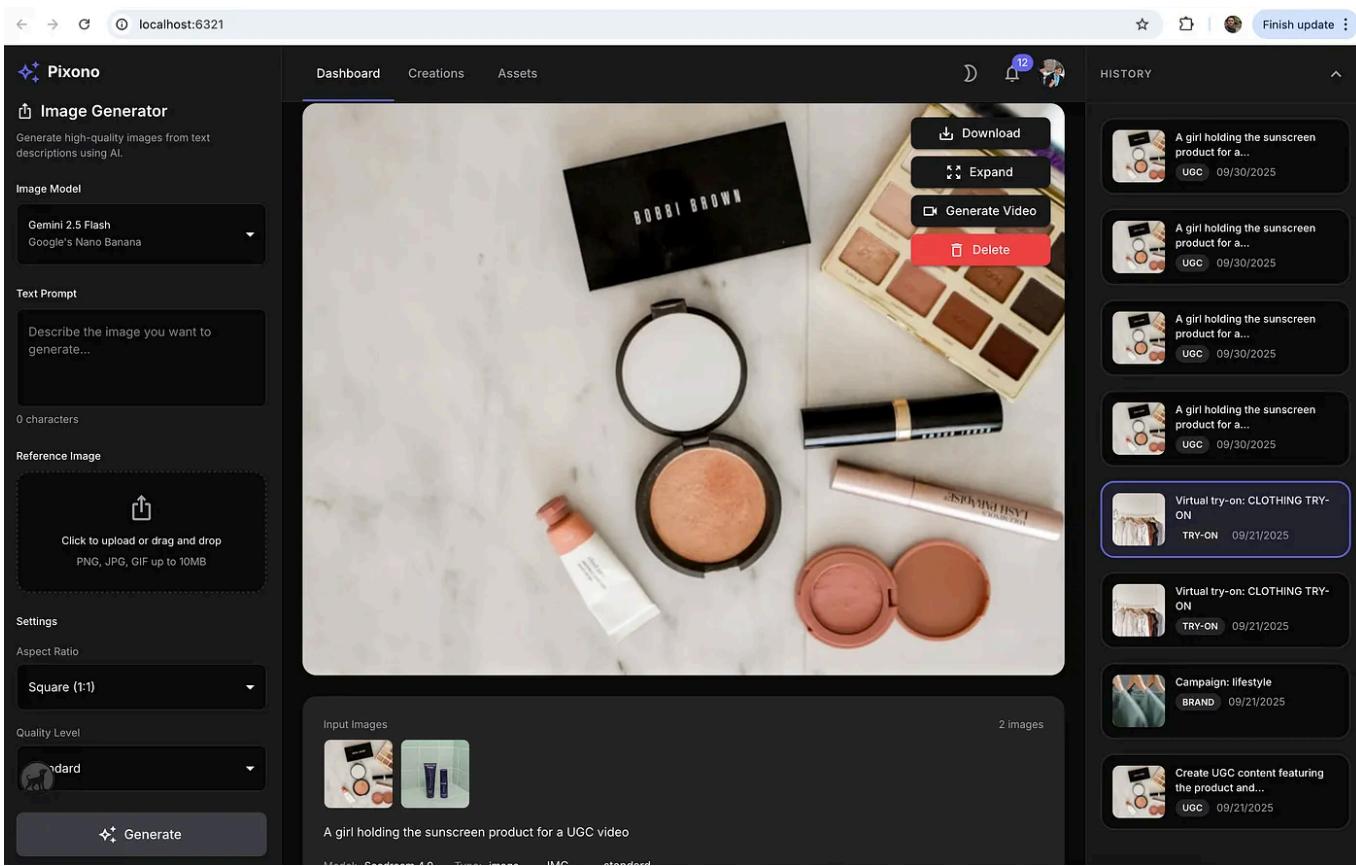
Right sidebar showing generation history with thumbnails, titles, and timestamps

This looks good to me, so I go ahead and click on the Approve Plan & Start Coding.



Plxono dashboard code generation in Kombai. Image by [Jim Clyde Monge](#)

After around five minutes, this is the UI dashboard that came up:



Pixono dashboard code generation in Kombai. Image by [Jim Clyde Monge](#)

This is super cool! I honestly did not expect the outcome to be very close to the reference image. Some small areas like the user avatar, header placement, and left tool navbar, but it's still very impressive.

If you know a website dashboard that you want to copy, just take a screenshot, feed it to Kombai, and let the coding agent do its magic.

If you examine the code and file structure, it's incredibly organized, and the build is successful on the first try.

The screenshot shows the Kombai IDE interface. On the left is a file tree for a project named 'KOMBAI'. The 'src' directory contains components like 'CanvasArea', 'Dashboard' (which includes 'Dashboard.tsx'), 'HistoryItemCard', 'HistoryPanel', 'ImageGeneratorForm', 'LeftSidebar', and 'TopNavigation'. It also includes files like 'dashboardMockData.ts', 'theme.ts', 'types.ts', 'enums.ts', 'schema.ts', 'formatters.ts', and 'App.dashboard.tsx'. A file named '# index.css' is selected in the tree. The main workspace shows the generated CSS code for '# index.css':

```
index.css – kombai
Dashboard.tsx 4 # index.css x
1 @import url('https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700');
2
3 :root {
4   font-synthesis: none;
5   text-rendering: optimizeLegibility;
6   -webkit-font-smoothing: antialiased;
7   -moz-osx-font-smoothing: grayscale;
8 }
9
10 * {
11   margin: 0;
12   padding: 0;
13   box-sizing: border-box;
14 }
15
16 body {
17   margin: 0;
18   font-family: 'Inter', 'Roboto', 'Helvetica', 'Arial', sans-serif;
19 }
20
21 #root {
22   margin: auto;
23   width: 100%;
24   height: 100vh;
25 }
```

Plxono dashboard code generation in Kombai. Image by [Jim Clyde Monge](#)

As someone who's been vibe coding for a couple of years now, it's annoying when code agents give you code that fails to build. That's a very basic feature, and I am glad that Kombai took care of it.

Kombai Pricing

Kombai offers a free plan that's perfect if you just want to explore its core features or use it for light, personal projects. It comes with 200 credits to start, plus an extra 300 credits as a limited-time bonus.

If you want to upgrade, here are the paid options:

Pricing

Free	Plus	Pro	Premium
Explore core features at no cost — perfect for light, personal projects.	Best suited for builders who want to occasionally use Kombai.	Best suited for builders who want to regularly use Kombai.	Best suited for builders who want to use Kombai heavily.
\$0	\$20/month	\$40/month	\$100/month
You get: ↗ 200 credits 缑 +300 credits Limited Offer	You get: ↗ 2,000 credits/ month	You get: ↗ 4,200 credits / month	You get: ↗ 11,000 credits / month

Kombai pricing page. Image by Jim Clyde Monge

- **Plus plan** at \$20 per month. It gives you 2,000 credits each month, which is more than enough if you're casually building projects or only using Kombai every now and then.
- **Pro plan** at \$40 per month. This one provides 4,200 credits every month and is best suited for developers who want to rely on Kombai as part of their daily workflow.
- **Premium plan** at \$100 per month. This tier is for heavy users who want maximum usage without worrying about hitting limits. It comes with 11,000 credits per month, making it ideal if you're building multiple apps or running bigger projects where you need consistent output from the agent.

Target Users of Kombai

Kombai isn't built for everyone. It's designed for people who already know their way around coding IDEs and want a serious tool to speed up frontend

work.

If you're a complete beginner with no coding background, you'll likely need to spend time going through the [guides](#) first. But if you're a developer who already works with VS Code, Cursor, Windsurf, or Trae, the learning curve is almost nonexistent.

The sweet spot for Kombai is developers who are building products regularly — indie hackers, startup engineers, and solo founders who can't afford to waste days writing the same boilerplate code again and again.

It's also useful for teams that want to move faster without sacrificing maintainable code. Instead of outsourcing templates or paying for pre-made boilerplates that never quite fit, you can generate something custom, readable, and ready to integrate.

Agencies might also find value here. If you're constantly building client dashboards and landing pages, Kombai can help you deliver projects faster.

Final Thoughts

If you're sick of paying tons of money for templates, boilerplates, or subscriptions to vibe coding tools that often end up producing broken code, you should definitely give Kombai a try.

The UI and UX are really good. For developers already familiar with VS Code, there's almost no learning curve at all. It feels fast and responsive, and the layout is simple and intuitive.

When it comes to the code itself, the quality is decent. The files are structured neatly, and the code is easy to read. One thing I really like is the planning phase before the build — it makes sure everything is aligned before generating the actual code. You can even tweak the color theme so the final result matches exactly what you have in mind.

Of course, I still recommend manually reviewing the codes generated by the coding agent before pushing anything to production.

Some may say that tools like Cursor or Lovable may already be enough, but take note that Kombai is an optimized coding agent for frontend development.

One reminder, though: Kombai is built into coding IDEs, so it's designed for advanced users. If you're just starting out or have no coding experience, I highly suggest going through the guides and examples first. They'll help you get familiar with how it works and avoid confusion when you start using it.



This story is published on [Generative AI](#). Connect with us on [LinkedIn](#) and follow [Zeniteq](#) to stay in the loop with the latest AI stories.

Subscribe to our [newsletter](#) and [YouTube](#) channel to stay updated with the latest news and updates on generative AI. Let's shape the future of AI together!

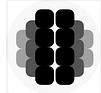
Technology

Kombai

Frontend

Web Development

AI



Published in Generative AI

60K followers · Last published 1 day ago

Follow

Stay updated with the latest news, research, and developments in the world of generative AI. We cover everything from AI model updates, comprehensive tutorials, and real-world applications to the broader impact of AI on society. Work with us: jimclydegm@gmail.com



Written by Jim Clyde Monge

27K followers · 2.6K following

Follow



4X Top Writer. Chief Editor at [generativeai.pub](#) | [zeniteq.com](#) | Work with me: jimclydegm@gmail.com

Responses (2)



Bgerby

What are your thoughts?



TechHarry

1 day ago

...

Impressive workflow! Kombai looks like a real time-saver for turning designs into frontend code. Thanks for

sharing your insights.

👏 1 [Reply](#)

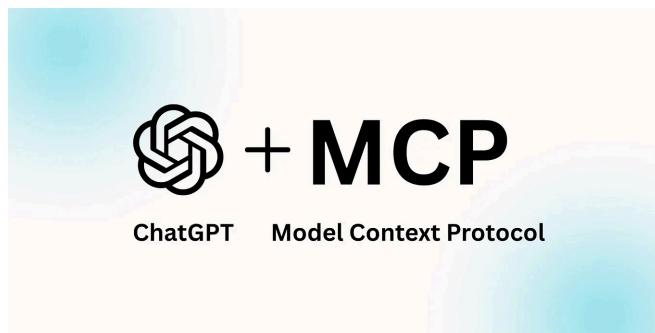
 Syed Munawar Hussain Bukhari
7 hours ago

...

Great post! Hope you'll check out mine too

👏 [Reply](#)

More from Jim Clyde Monge and Generative AI



 In Generative AI by Jim Clyde Monge 

ChatGPT Now Supports MCPs

OpenAI introduces developer mode in ChatGPT, enabling MCP server tools.

⭐ Sep 12  507  6



...



 In Generative AI by 302.AI

Official Nano-Banana Prompting Guide with Test Case Comparisons

Unlock Google's Official Prompt Tips with 6 wild cases to instantly Improve Your Image...

⭐ Sep 4  188  5



...



Hierarchical Reasoning Model (HRM)

A tiny brain that
embarrasses
giant LLMs



In Generative AI by Dr. GenAI

Hierarchical Reasoning Model (HRM): a tiny brain that...

HRM is a 27M-parameter model trained in hours that beats giant LLMs on reasoning...



Sep 7

306

6



...



Cursor



Warp



In Generative AI by Jim Clyde Monge

Why I Switched From Cursor to Warp Code

I've tried a handful of vibe coding platforms to find the best replacement for Cursor AI.



Sep 5

243

9



...

See all from Jim Clyde Monge

See all from Generative AI

Recommended from Medium



 In Level Up Coding by Yasar Sandeepa

Google's New AI – Nano Banana: Top 10 Use Cases That Will Blow...

How Nano Banana is redefining the future of AI Image Gen

⭐ 6d ago ⌘ 195 💬 2



...



 Jaytech

11 Free Microsoft AI Tools You Should Know in 2025 🤖

Photo by BolivialInteligente on Unsplash

⭐ Oct 1 ⌘ 19 💬 6



...



 In The Generator by Jim the AI Whisperer

The words “blah blah blah” increase AI accuracy

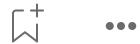
Who needs Chain of Thought when “blah blah blah” works?

 In Data Science Collective by Erdogan T

Build Your Private Language Model: Local and Specialized For...

A complete step-by-step guide from setup to deployment of local language models, makin...

⭐ 6d ago ⌘ 644 💬 4



...



 Joe Njenga

I Tested Newly Released GLM 4.6 (And Discovered a Cheaper Way t...

GLM 4.6 has been released just a few days after Claude 4.5, and it's another coding...

⭐ 2d ago ⌘ 39 💬 1



...

Welcome to BitNet.
How can I help you today?

What do you want to know?

CPU

By messaging BitNet, you agree to our Terms and Privacy Policy.

 In Coding Nexus by Algo Insights

Microsoft Just Declared War on the GPU Mafia: Meet bitnet.cpp

Bitnet.cpp will break the GPU Lock

 5d ago  3.5K  73

 4d ago  244  3

[See more recommendations](#)