

PROJECT INTRODUCTION

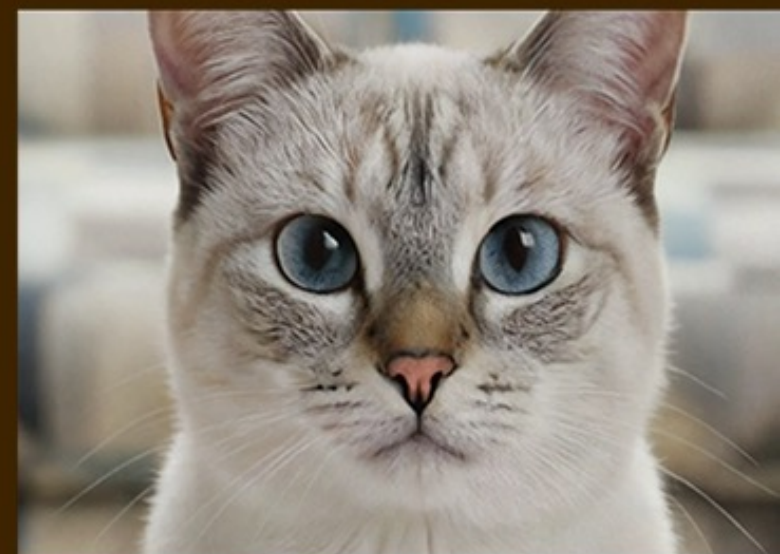
My project is Prrrfct, a **virtual pet companion that offers a calmer alternative to social media.**

LEARNING OPPORTUNITY

I'm **exploring AI workflows in product creation** to reinvent the virtual pet category in the \$158B pet market.*

*The American Pet Products Association.

Prrrfct



CASE STUDY INTRODUCTION

In early 2025, I created **a project to apply what I was learning about AI product design, research, and development** through courses from DeepLearning.AI, IBM, Coursera, and other educational platforms.



Professional Certificate in AI for Accessible User Experience Design

Stanmore School of Business

Issued Apr 2025

Credential ID 8c448ec5-96fa-4b6b-8764-c45ef10bd553

[See credential](#)



AI For Everyone

DeepLearning.AI

Issued Mar 2025

Credential ID WVZ6UG65H3FA

[See credential](#)



Agentic AI Fundamentals: Architectures, Frameworks, and Applications

LinkedIn

Issued Mar 2025

[See credential](#)



GenAI for UX Designers

Coursera Instructor Network

Issued Mar 2025

Credential ID TK2UXOL47FA7

[See credential](#)



Generative AI: Introduction and Applications

IBM

Issued Mar 2025

Credential ID VVSQTO7EP67J

[See credential](#)

Certifications from the first AI classes in my learning journey, which continues today

MY CHALLENGE

Create a repeatable, scalable AI design practice using a new product idea as the proof-of-concept (POC).

Pain points

1. Evaluate AI platforms as they evolve.
2. Build technical skills to port a Figma design system into React and launch a beta site.
3. Address code generation drift.

My solutions

1. Keep the process flexible to use the best tool as platforms evolve.
2. Keep code platform-agnostic.
3. Replacing my custom skill and harness work with community techniques.

PROJECT OVERVIEW

I'm using an **iterative product design process**, which evolves as I drive further development.

AI Design Phases

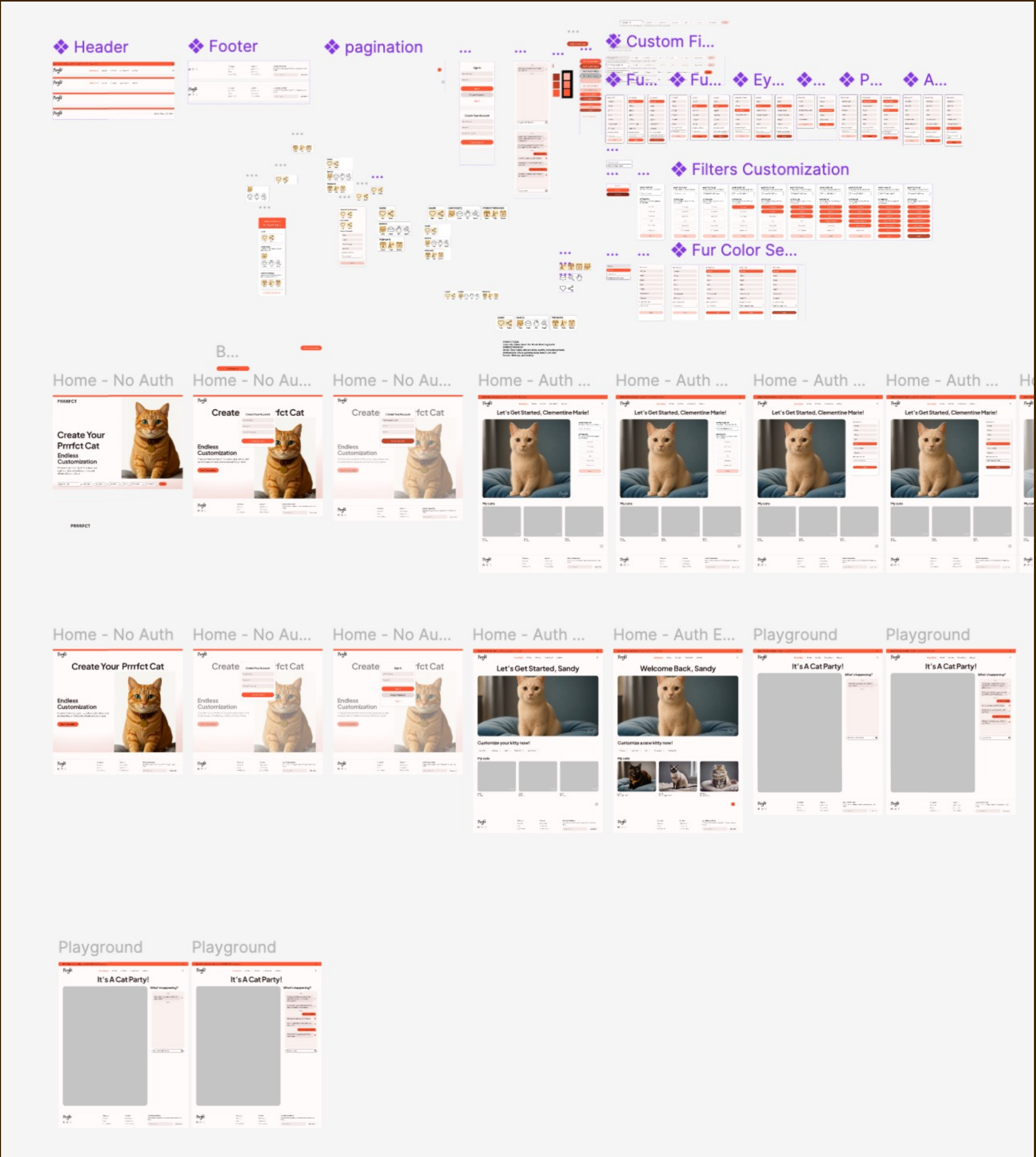
- 1. Exploration
- 2. Direction
- 3. Standardization
- 4. System design
- 5. Production
- 6. Validation

Timeline & Team

Just me and the AI army.
(ChatGPT, Cursor, Claude Code, Bolt, v0, Lovable, etc.).
Early concepting began in 2025;
the app is slated to beta in 2027.

Results

I launched a sign-up site in late 2026 to as an AI process POC.
I'm now actively developing the mobile-first app.



The process involved visual and interaction design, component creation, asset generation, etc.

PHASE No. 1

Exploration

How can AI aid discovery?

- Careful prompting results in efficient competitive and comparative research findings.
- Research findings provide the basis for Product Requirement Documents (PRDs).

Challenges

1. It’s tricky to know which articles, posts, and videos to consume and trust in research activities.
2. Creating a PRD from scratch is daunting and time consuming.

Learnings

1. Prompting AI to provide sourcing with research narrows useful options.
2. AI can synthesize research findings to start the PRD writing process.

Insights

AI speeds up research but double-checking resources ensures information quality. Research findings inform the PRD writing process.

2025 U.S. pet market snapshot

Americans spent about **\$158 billion on pets in 2025**, up roughly **3.7% from 2024**, according to APPA's subsequent State of the Industry reporting. APPA projects the market at roughly \$165 billion for 2026.  +1

[American Pet Products Association market data ↗](#)

The biggest spending pools remain food/treats, veterinary care, supplies/OTC products, and services. One important change is that growth is becoming more volume-driven again rather than simply reflecting inflation.  +1

At the same time, consumers are more price-conscious. In a January 2025 Packaged Facts survey, **31% of pet owners said inflation had hurt their ability to buy**

nonessential pet products. Pet owners treat pets as family, but pet insurance delivers a meaningful benefit.

2. Cat-first products

This may be one of the better white-space directions.

A lot of pet innovation is essentially:

"We built something for dogs...and there's also a cat version."

I'd invert that.

Interesting problems include hydration, urinary health, hairballs/digestion, litter odor, enrichment for indoor cats, stress/anxiety and weight control.

A particularly compelling concept could be a **cat hydration platform**—for example, functional broth/topper concentrates with hydration plus targeted benefits.

Yes — I think a **modern virtual cat** could be a more interesting consumer-product opportunity than another physical pet accessory.

The important distinction is that I wouldn't build "Tamagotchi, but it's a cat." I'd build a **virtual cat that actually develops a personality and relationship with you**.

Tamagotchi itself is still commercially active: Bandai launched Tamagotchi Paradise in 2025 with color screens, breeding/genetics, multiple environments and 50,000+ possible character combinations.  +1 Tamagotchi was also inducted into the World Video Game Hall of Fame in 2025, which is a useful signal that the underlying mechanic has real cultural durability. 

Refining prompts during research sessions speeds up the process

PHASE No. 2

Direction

How can I move beyond generic AI output?

- Establish/reinforce project context.
- Define desired outcomes (not methods) in a PRD.
- Create wireframes appropriate to needs.

Challenges

1. Context statements are static and a PRD may impose unintended limitations.
2. Input and output quality are linked.

Learnings

1. Update (or even radically change) context statements and PRD as work progresses.
2. Use a cost/benefits matrix to determine a fidelity level, which may change over time.

Insights

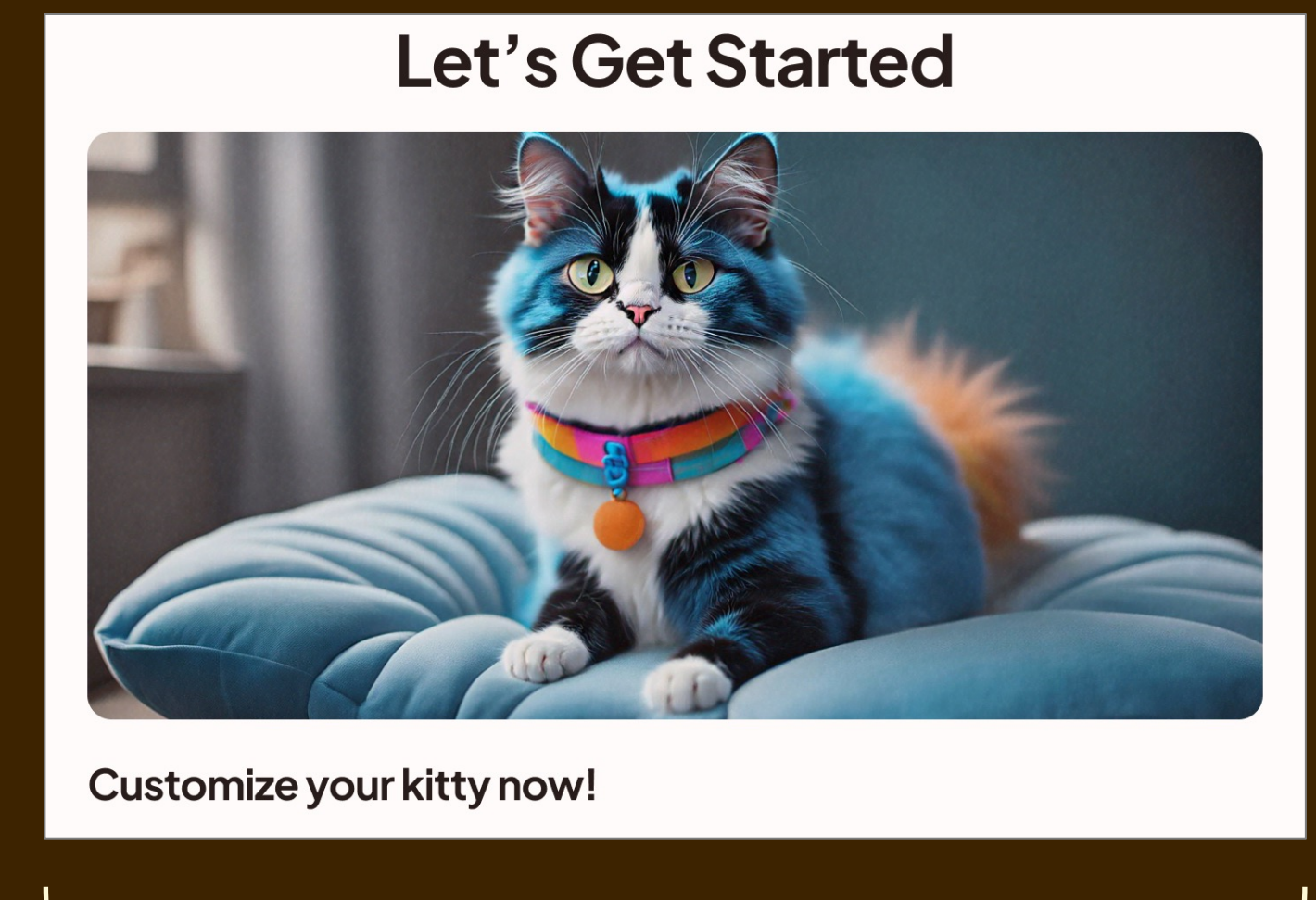
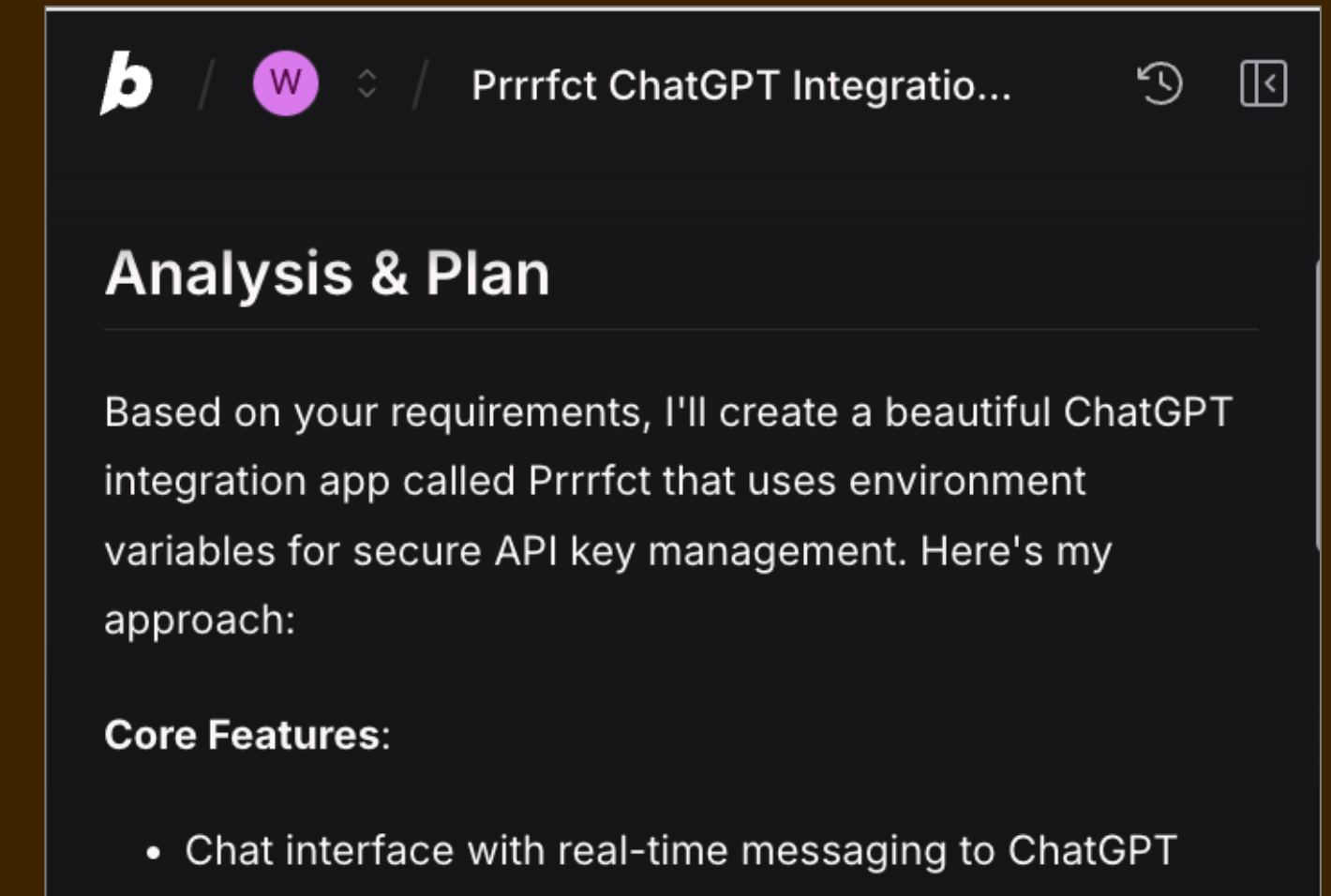
Establishing context from the start is important, but flexibility in updating it is essential.

Recognizing and responding to your own preconceptions as the process continues is key to improving outputs.

PRD prompt

↓
AI-generated plan

↓
Output improving... slowly



There were clearly some issue around character styling!

PHASE No. 3

Standardization

How can I create standards to ensure consistent outputs with less corrections?

- Write standardized prompts with positive examples.
- Include constraints and negative examples.
- Manage context windows for session success.

Challenges

1. Few prompt examples have been shared as guidance.
2. Examples are easy to find; constraints are difficult to devise.
3. Overloaded context windows can destroy session work.

Learnings

1. Save standards and paste them into all prompts.
2. Create one image with both positive and negative examples.
3. Streamlined standards use context windows efficiently.

Insights

Creating textual and visual standards increases consistent output and context window efficiency, preventing prompt session disasters (a lesson hard learned!).

Standards
codified



Standards
applied

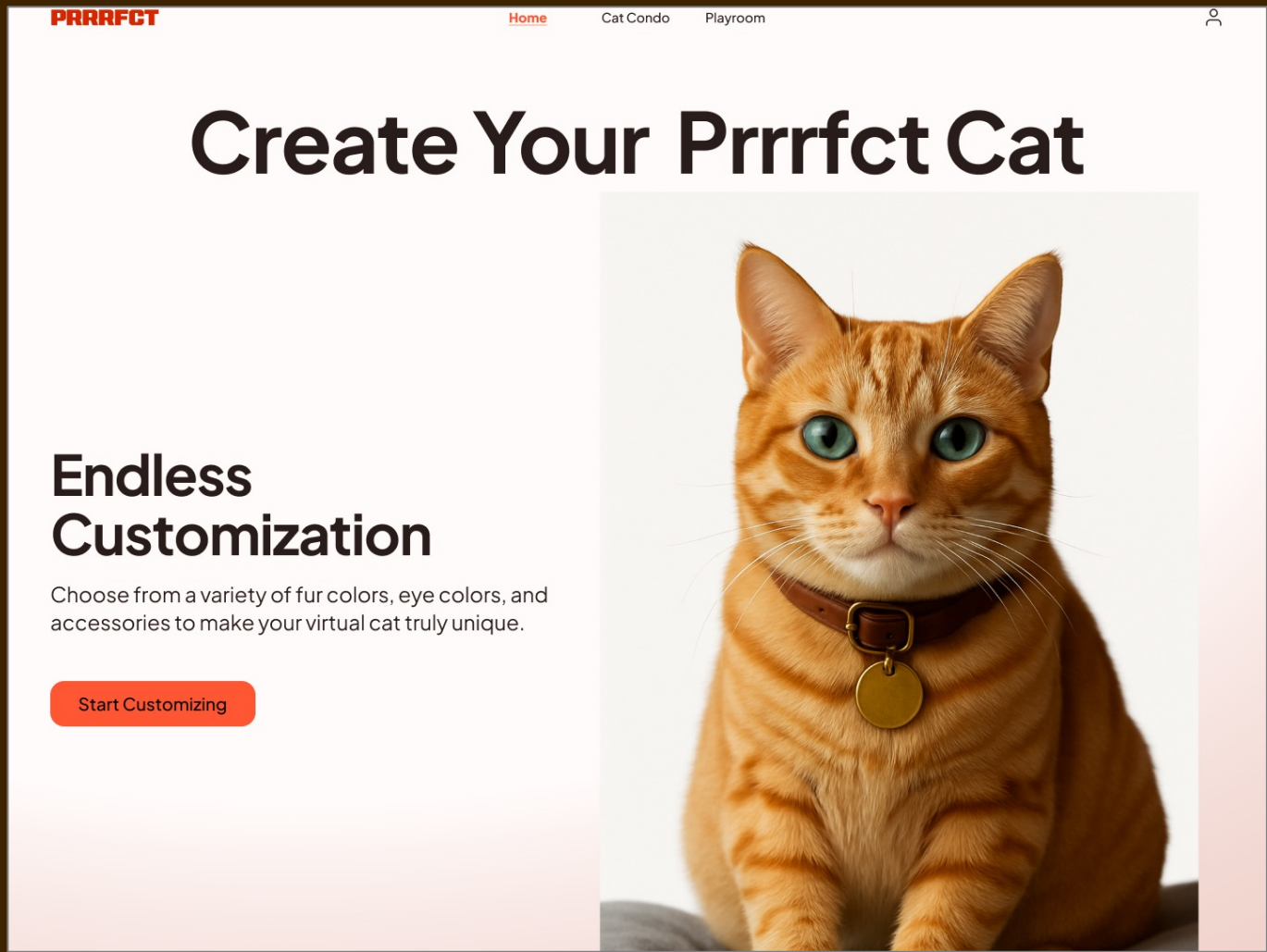


More consistent
output

Cat creation prompt:

A realistic, high-quality portrait of an **orange tabby** sitting with the **entire cat and entire cushion fully visible**. The cat should be **medium** in size, centered on the cushion, with **extra space on the top and bottom** of the image, **approximately 25% of the total image width on each side**. Use a clean, seamless **white background** with no visible textures or patterns. Add a **soft, diffused studio-lit shadow directly beneath the cat** to enhance realism without suggesting a specific surface.

The cat should face forward, looking directly at the camera. The lighting is **soft and diffused**, evenly illuminating the cat and its shadow. The cat should have **lifelike fur texture, natural coloration, and accurate anatomy**.



The cat creation standards allowed me to create cats with similar visual stylings over multiple sessions

PHASE No. 4

System design

How can I turn my standards into a scalable, reusable system?

- Utilize tokens in the Figma design system.
- Create reusable specification harnesses (constraints).
- Ensure system success by managing context windows.

Challenges

1. Porting designs from Figma to React will expose weaknesses.
2. AI's success in remembering harnesses is uneven.
3. Context windows fail without warning and work can be lost.

Learnings

1. Consistent token exports are key to success in React.
2. Devise a system for saving harnesses locally for reuse.
3. Rely on single-task prompt sessions to manage context.

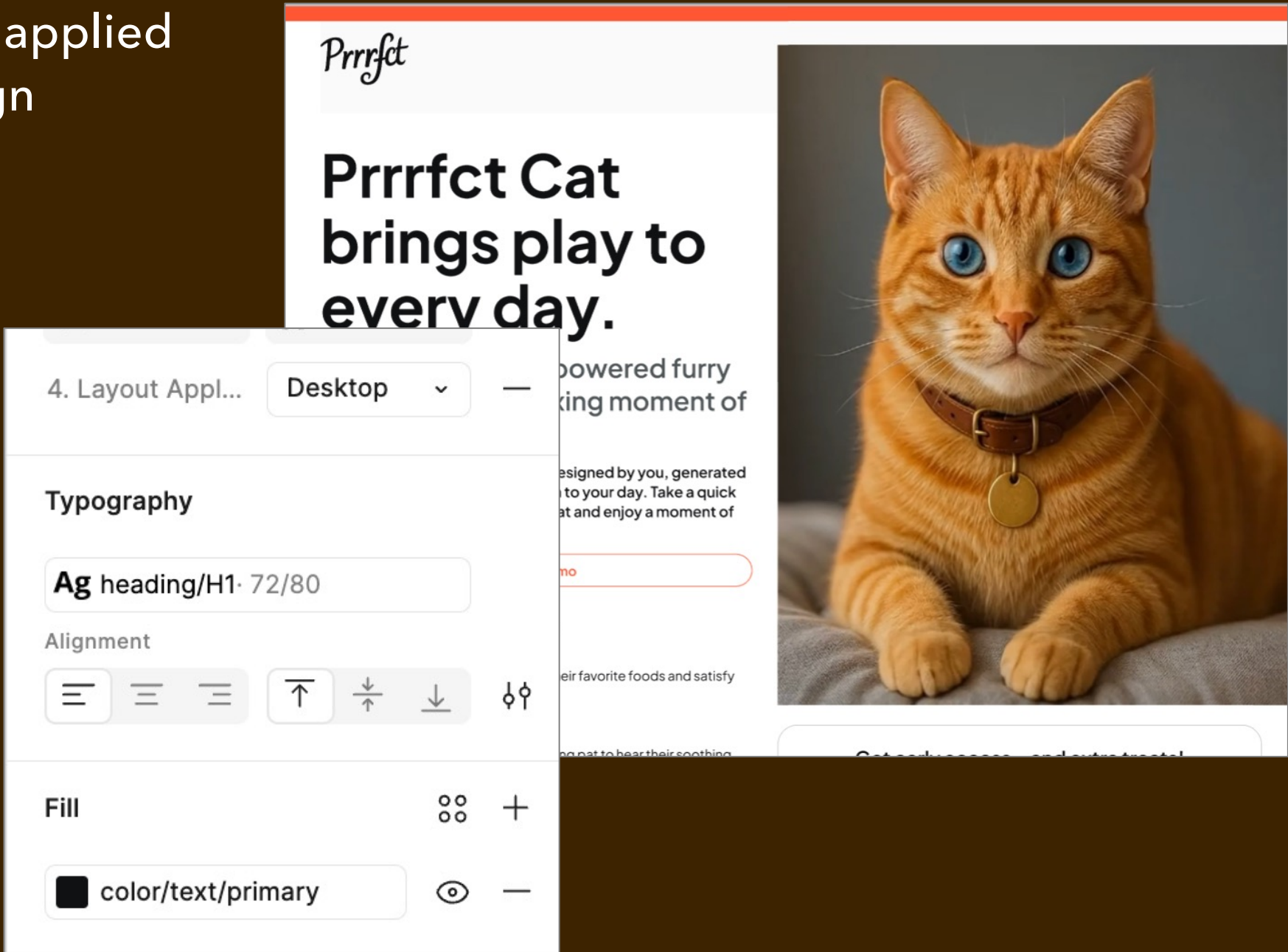
Insights

Carefully applied tokens translate smoothly into React code. Reusing harnesses in single-task sessions improves workflow and context management efficiency.

Tokens defined in Figma



Tokens applied in design



PHASE No. 5

Production

How do I port a Figma design system into a coded design system?

- Design-to-code workflows facilitate the transition.
- Transparent code architecture aids understanding and troubleshooting.
- A development environment (IDE) bridges the transition.

Challenges

1. Design-to-code is a nascent concept without much support.
2. Vibe coding platforms optimize speed over transparency.
3. Using an IDE entailed a learning curve for me.

Learnings

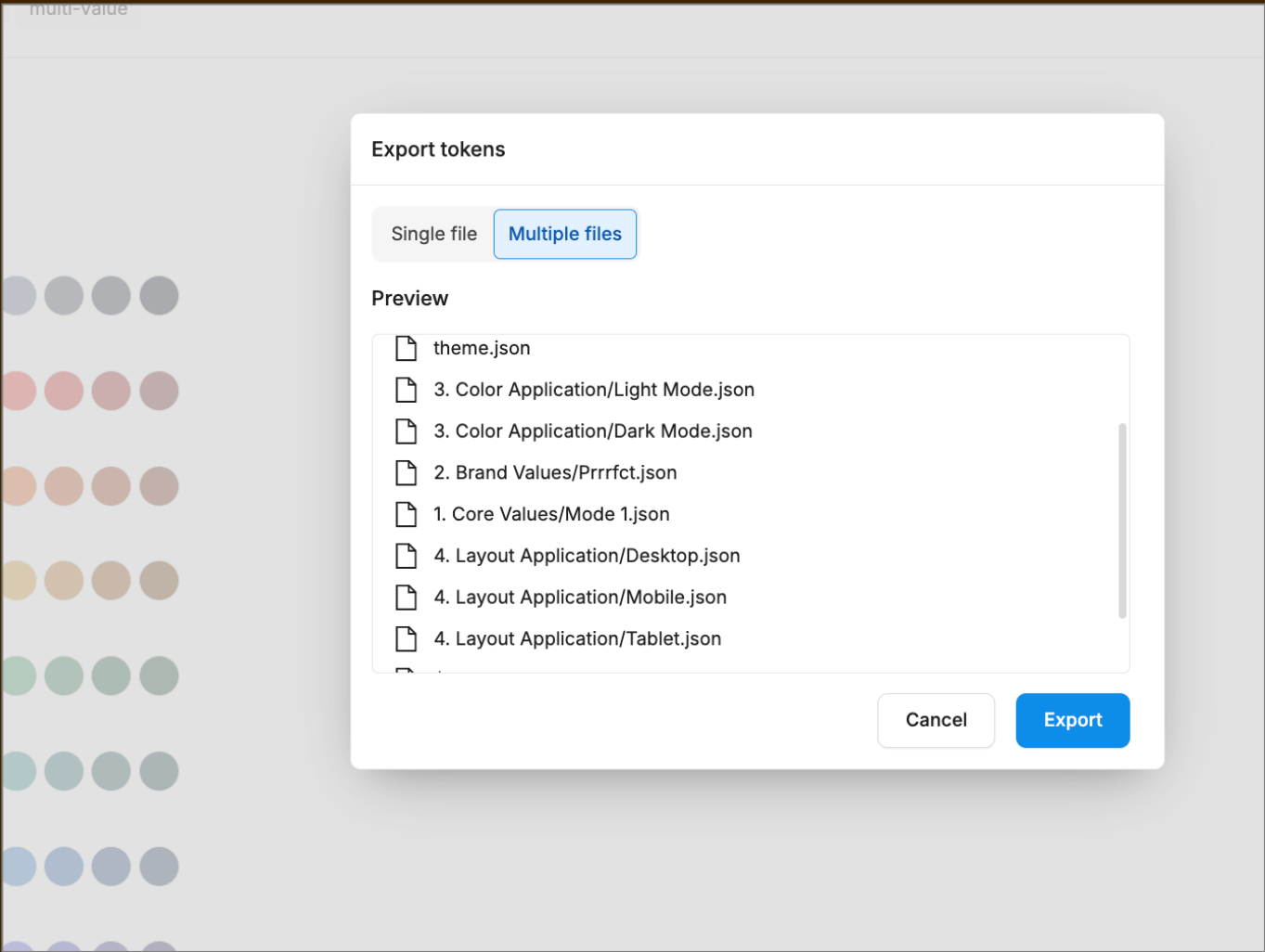
1. Created my own workflows through trial and error.
2. Vibe coding tools don't offer enough visibility or control.
3. Cursor + React is compatible with AI tools and design systems.

Insights

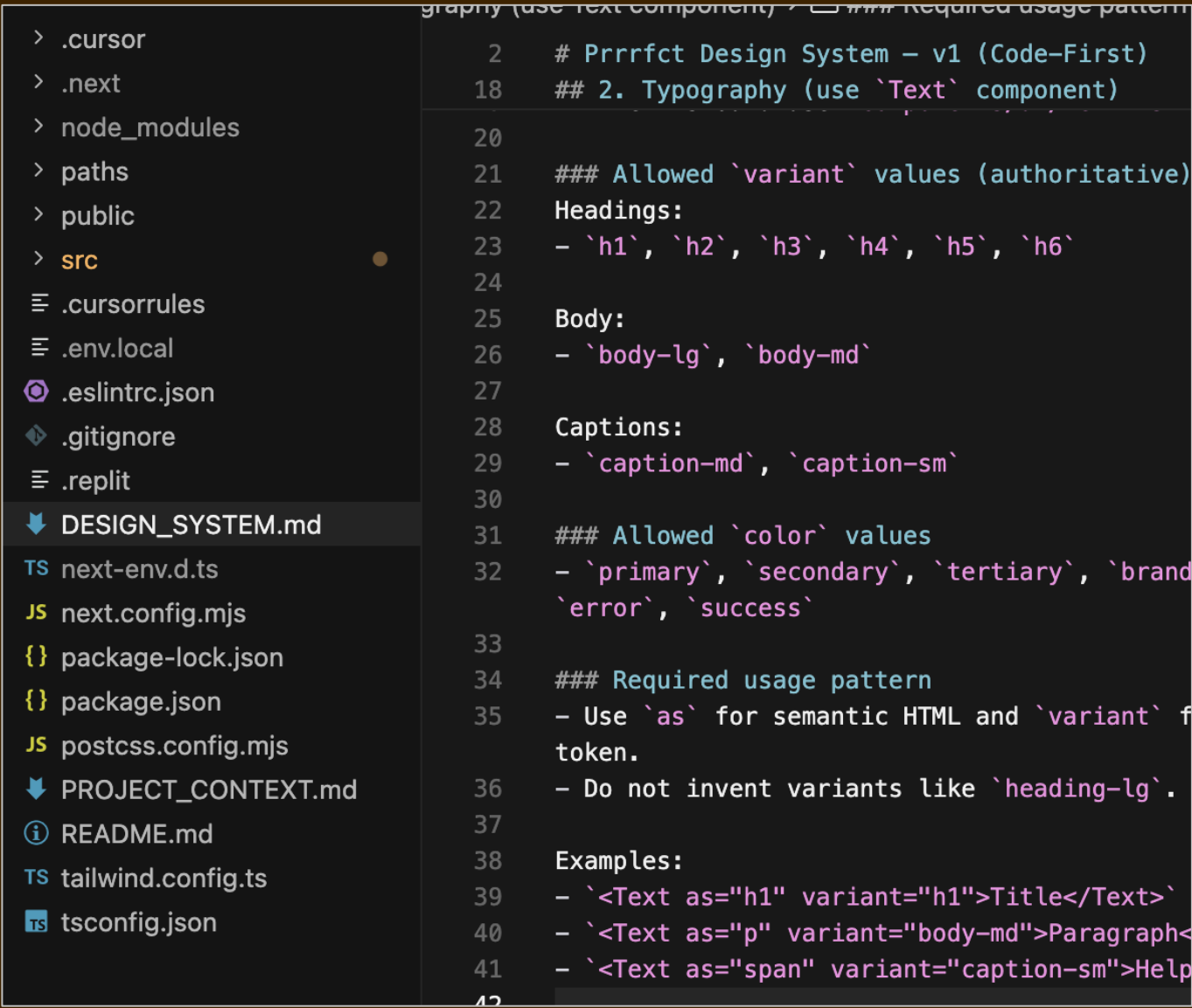
In Sept. 2025, I had to pioneer original workflows, understand code architecture, and learn to use IDE's for code generation.

NOTE: Some tech references are outdated by now.

Tokens
exported to
JSON



JSON imported;
code
architecture
easy to see and
edit in an IDE



PHASE No. 6

Validation

How can I effectively evaluate and maintain the system?

- The QA process will lead to more code iteration.
- Troubleshooting will require code iteration.
- Evolving community practices have defined more efficient design-to-code workflows.

Challenges

1. An LLM can get stuck in repetitive, non-productive QA loops.
2. An LLM may have difficulty troubleshooting its own code.
3. My workflows were short-term solutions.

Learnings

1. Using multiple LLMs to evaluate and troubleshoot code minimizes endless loops and arrives at solutions more quickly.
2. The wider community is creating and documenting new, improved workflows

Insights

Multi-tool QA evaluations/troubleshooting is efficient, and a community strength is creating shared processes in real time.

A secondary LLM reviewed the code and surfaced issues the authoring LLM had overlooked

Detailed Feedback on `Button.tsx`:

✔ What's Good:

1. **TypeScript props interface** - Well-defined types
2. **Variant system** - Good pattern for different button styles
3. **Accessible attributes** - Spreading `...props` allows aria-labels, etc.
4. **Disabled state handling** - Good UX consideration
5. **Design token usage** - Using your custom Tailwind tokens

✖ Critical Issues:

1. Remove unnecessary React import

```
typescript
// ✖ Remove this
import React from 'react';

// ✔ This is all you need
interface ButtonProps extends React.ButtonHTMLAttributes<HTMLButto
  // ...
}
```

2. Focus ring color issues

```
typescript
// ✖ Current - generic focus ring
focus:ring-2 focus:ring-offset-2

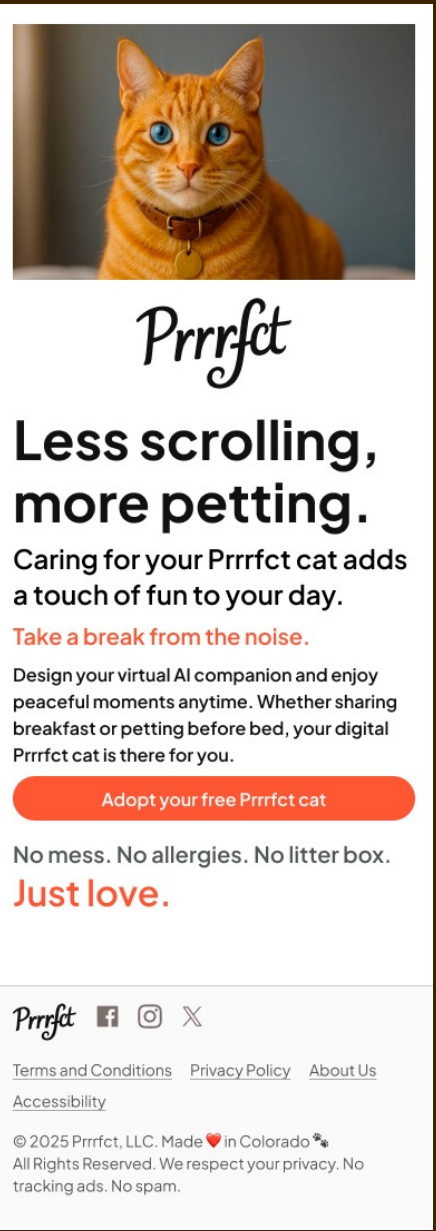
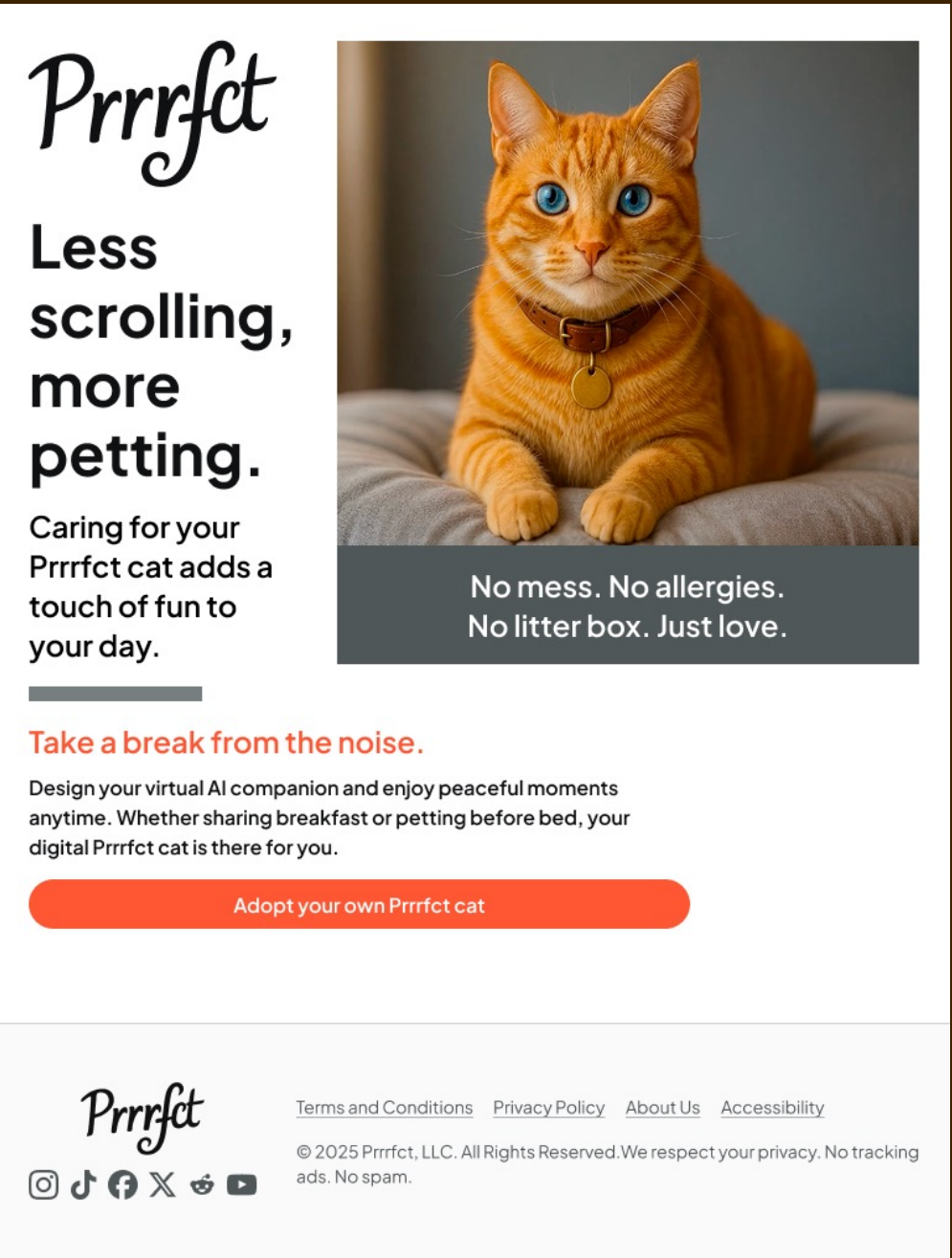
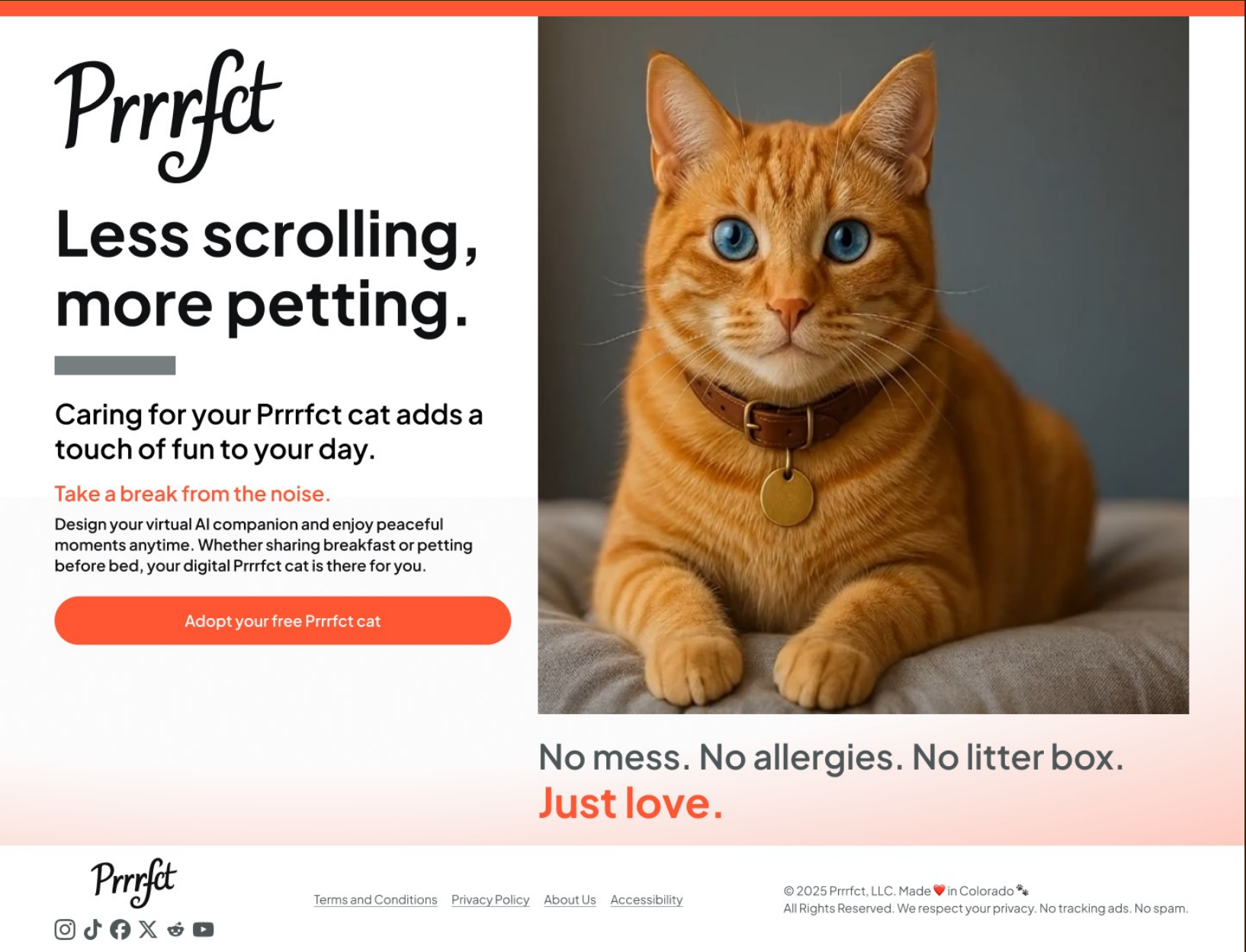
// ✔ Better - specify color for accessibility
focus:ring-2 focus:ring-brand-primary focus:ring-offset-2
```

Summary

Summary

Through Prrrfct, I developed an **AI-native product design practice that combines product strategy, reusable design systems, design-to-code workflows, and AI-assisted development.**

I utilize a platform-agnostic framework that adapts as the technology continues to evolve.



Desktop, tablet, and mobile designs

Next steps

Continuing the iterative product design process, I'm **converting my workflow files into better-structured markdown files**.

I'm also comparing the latest updates to agentic coding assistants for continued development.

About me

As a Senior Product Designer, **I transform complex ideas into scalable, human-centered experiences** using product strategy, design systems, and AI workflows. **I bridge UX strategy and front-end execution** to deliver real-world impact.

