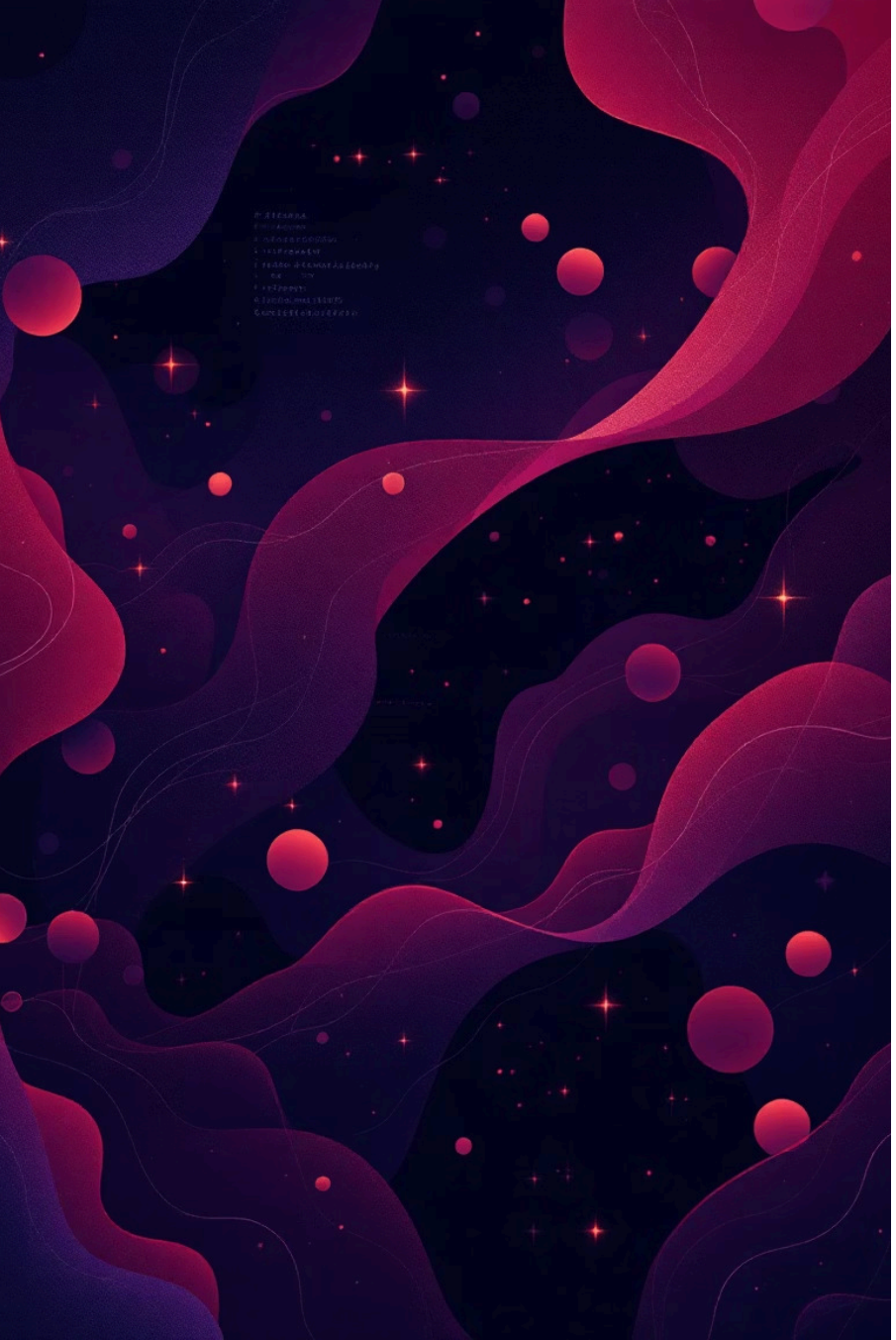


FigmAI: Transforming Design to Code with Generative AI

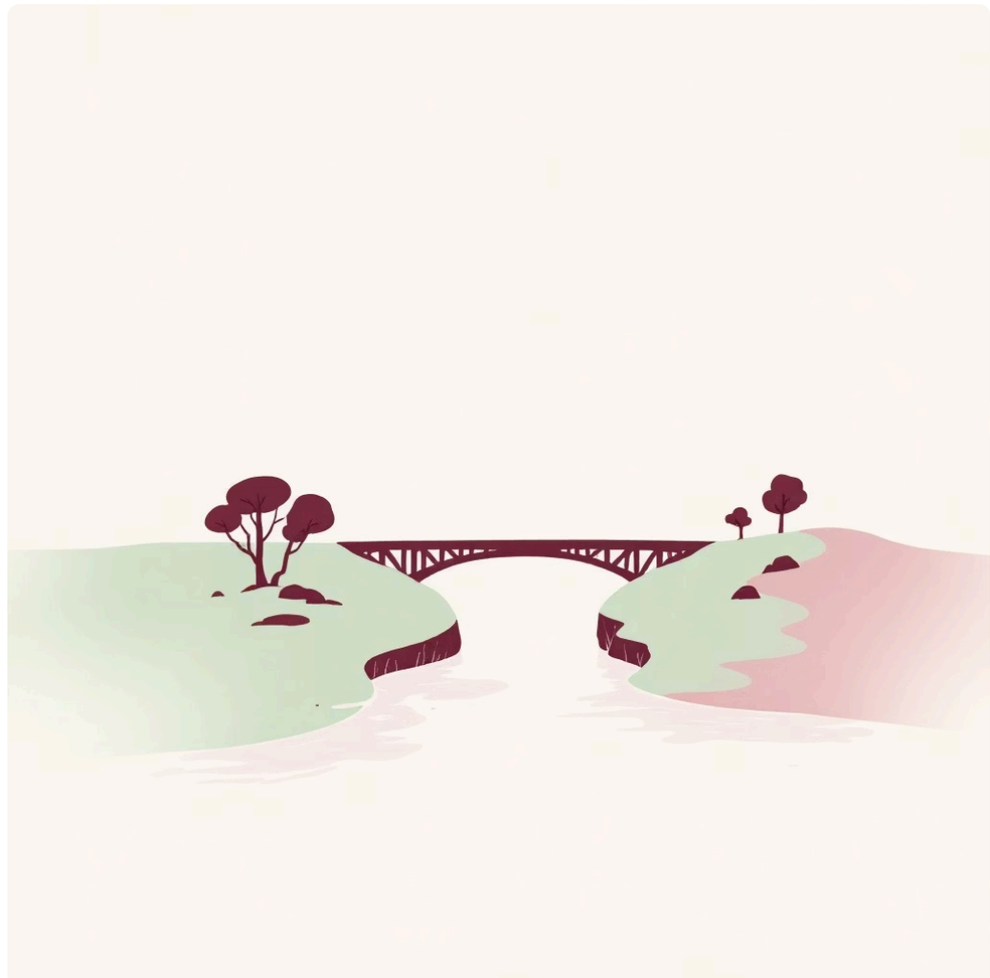
Bridging the gap between design and development with intelligent agents

The Design-to-Code Challenge

Current Landscape

Designers and developers work in separate ecosystems, leading to:

- Inconsistent implementation of designs
- Lengthy development cycles
- Communication breakdowns between teams
- Increased technical debt over time



Our solution transforms this fragmented workflow into a seamless process, reducing friction and accelerating production.

FigmAI: Solution Overview

A powerful AI-driven platform that transforms Figma designs into production-ready React code

Figma API Integration

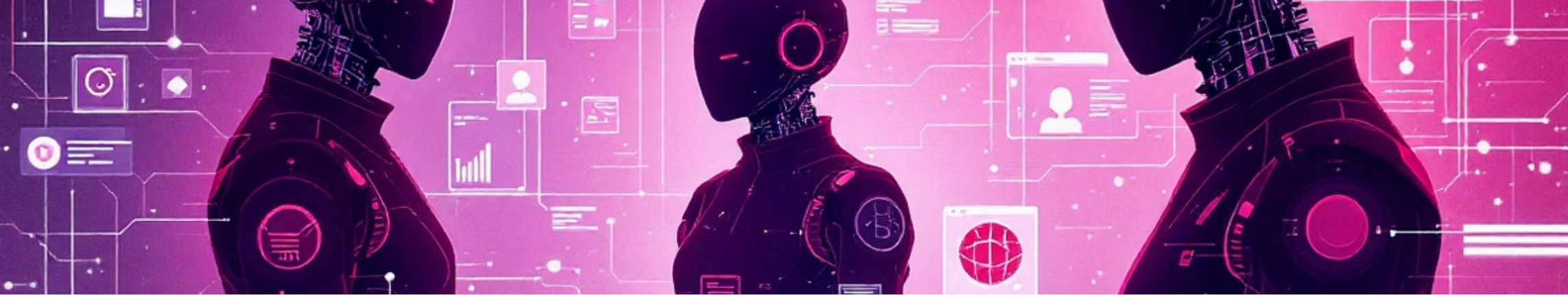
Connects to your Figma account and extracts component designs, preserving all styling details and relationships

Multi-Agent Processing

Processes designs through specialized AI agents that understand design patterns and translate them to code

React Component Output

Generates high-quality, maintainable React components that faithfully represent the original design



The Three-Agent Architecture

1

Figma Analysis Agent

Specialized in parsing and understanding Figma's complex component structures, layer hierarchies, and design systems

- Maps visual elements to semantic structures
- Identifies reusable patterns and components
- Preserves design intention through metadata analysis

2

Code Generation Agent

Transforms design specifications into clean, optimized React code following modern development practices

- Implements responsive behavior
- Maintains accessibility standards
- Creates component props and variants

3

Testing & Refinement Agent

Validates code output and refines based on human feedback to ensure quality and accuracy

- Performs visual regression testing
- Implements feedback in subsequent iterations
- Ensures cross-browser compatibility

Proof of Concept Results

Key Achievements

Component Fidelity

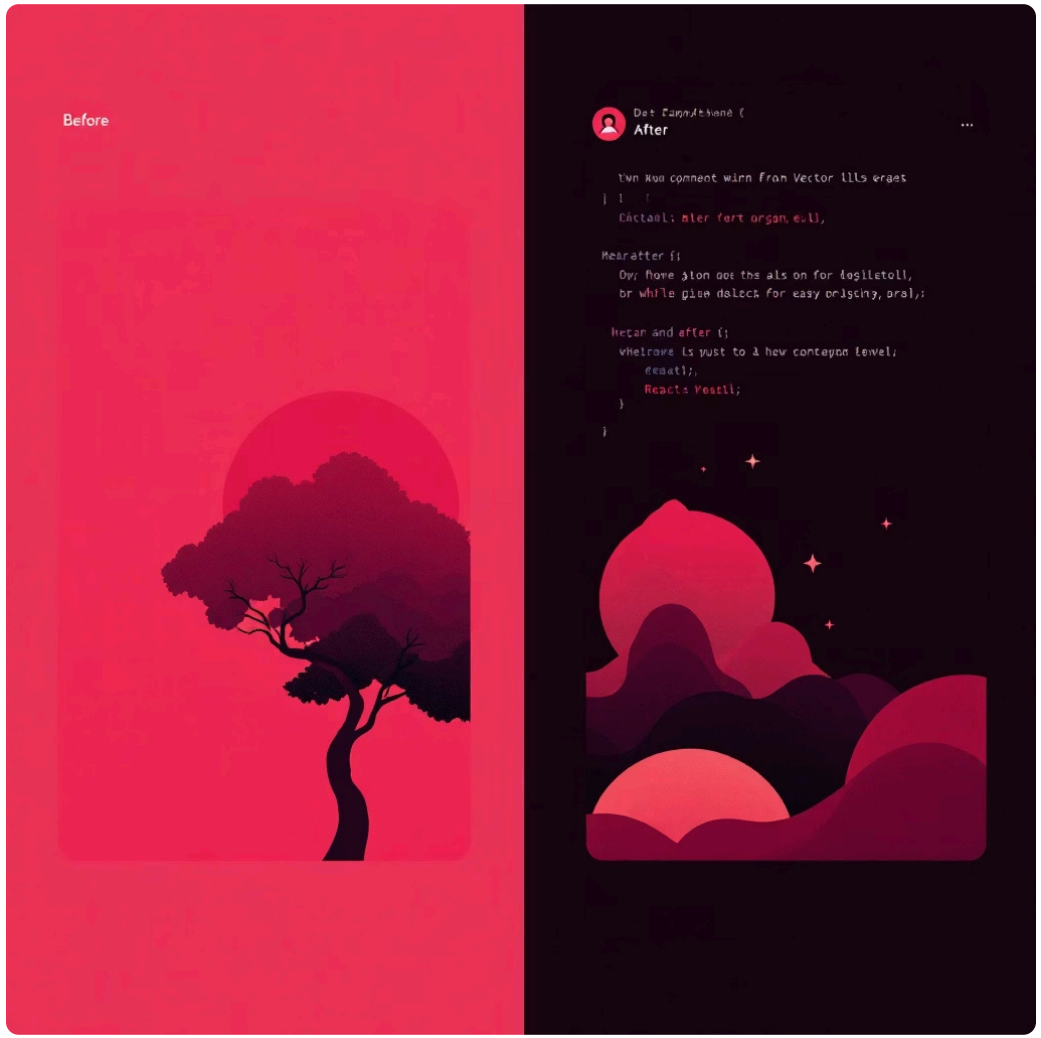
Successfully translated complex Figma components with 95% visual accuracy

Code Quality

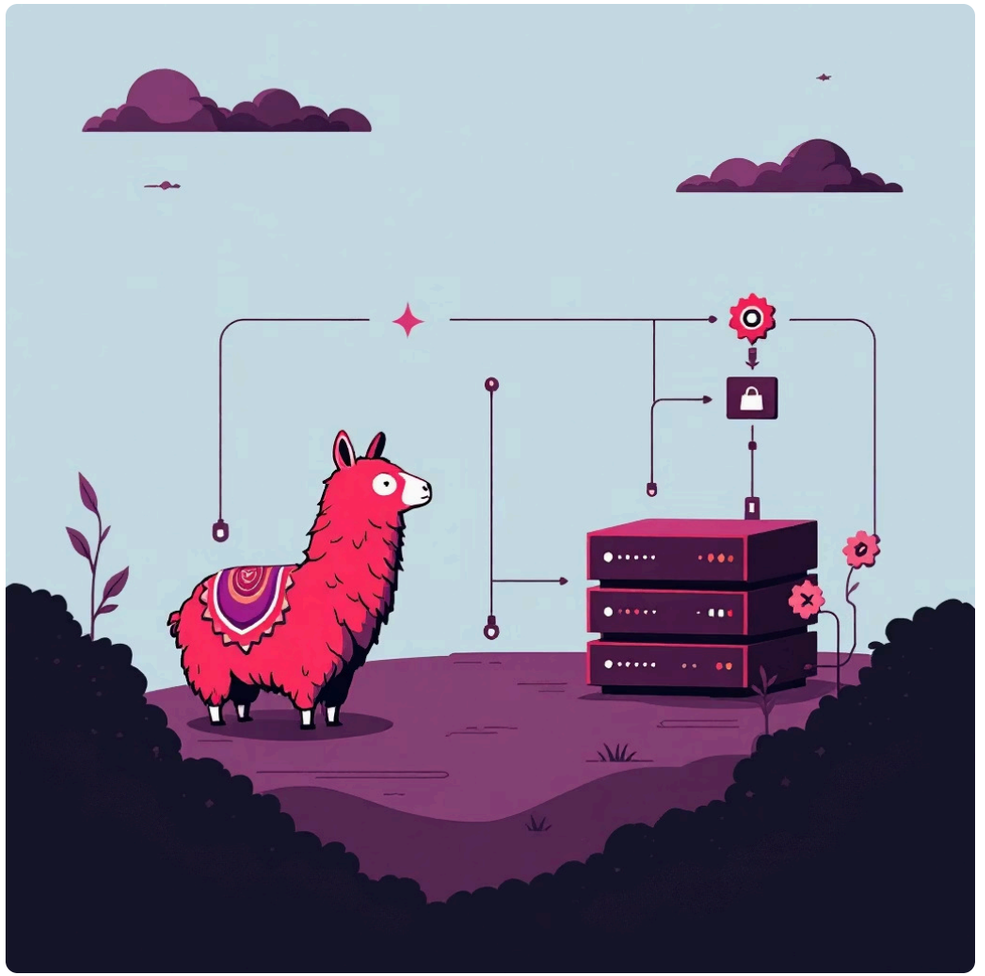
Generated clean, well-structured React code that passed standard linting and best practices

Developer Efficiency

Reduced implementation time by 70% compared to manual coding



Current Technical Implementation



Local LLM Implementation

- Currently powered by Ollama running locally
- Complete end-to-end workflow demonstrated
- Processing time: 3-5 minutes per component

Technical Requirements

- Figma API access for component extraction
- React development environment
- Sufficient local compute resources

Current performance is limited by local compute constraints, with significant potential for optimization with cloud-based LLMs.

Current Status & Roadmap

What We've Done

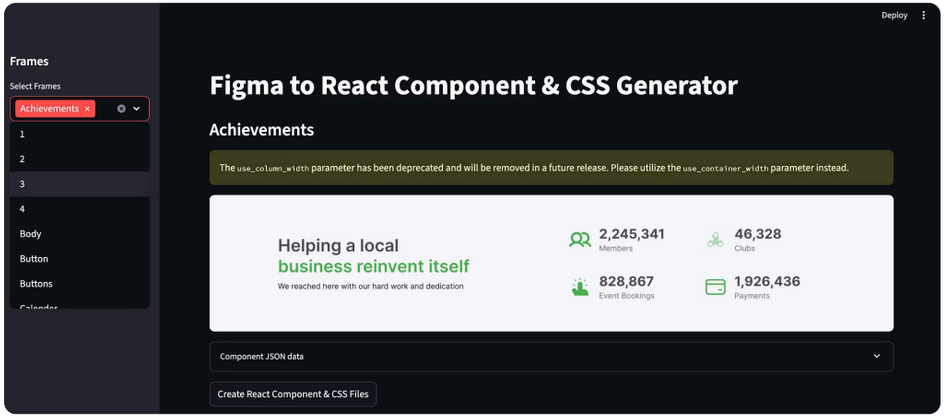
- **Local LLM Implementation:** Successfully integrated Ollama for local AI processing.
- **End-to-End Workflow:** Demonstrated a complete pipeline from Figma design to React code.
- **Initial Performance:** Currently processing components in 3-5 minutes.
- **Key Features:** Figma API integration, multi-agent processing, and React component output are functional.

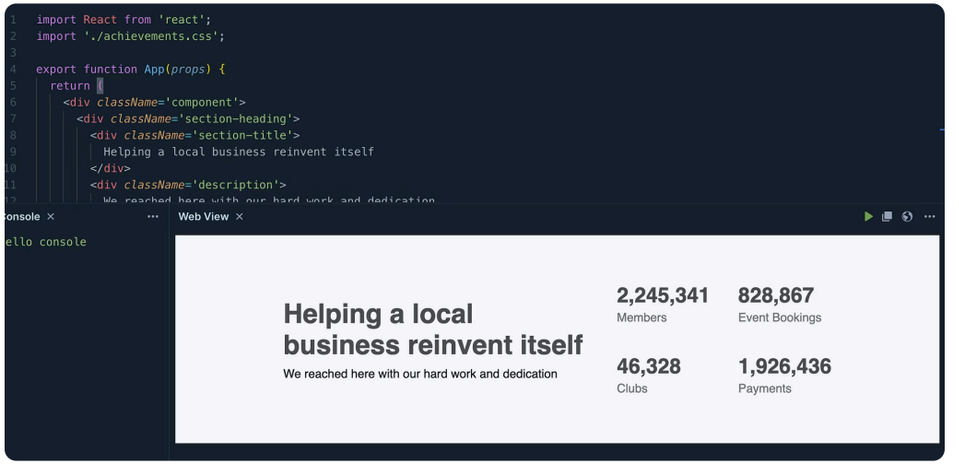
What's Next

- **Cloud LLM Integration** Transition to high-performance cloud LLMs to reduce generation time to under 30 seconds and enable parallel processing.
- **Component Library Integration** Implement automatic publishing to Storybook, integration with npm registry, and robust version control.
- **Advanced Features** Add support for animations and interactions, state management integration, and custom hook generation.
- **Enterprise Deployment** Develop team collaboration features, enforce design system consistency, and integrate with CI/CD pipelines.

FigmAI in Action: Design to Code Visuals

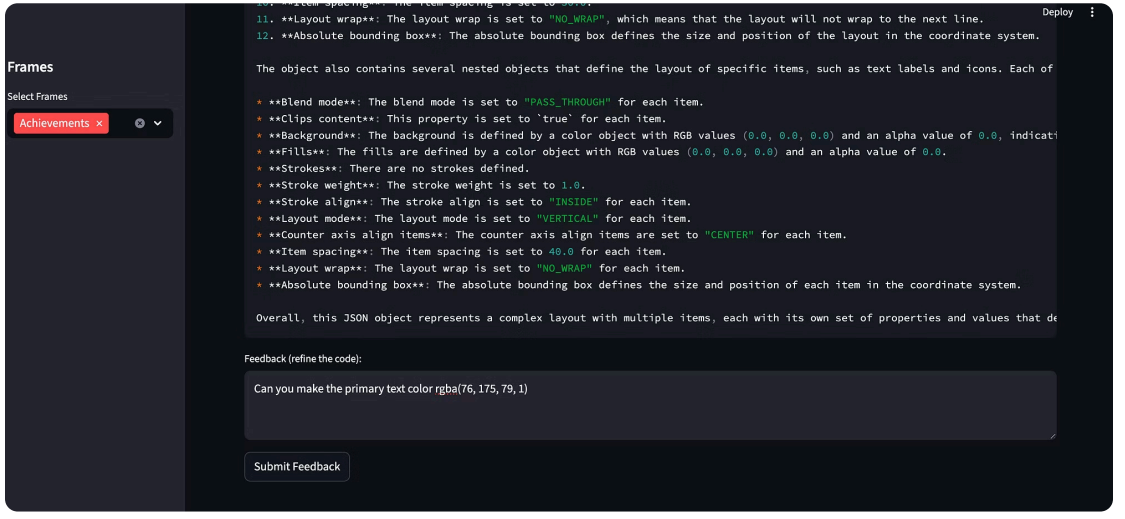
See how FigmAI translates designs into clean, functional React code with high fidelity.





These visuals highlight the seamless transition from design concept in Figma, through our intelligent processing, to the final, production-ready React component.

Iterative Refinement: Your Feedback, Our AI's Improvement



Refine & Repeat: Continue providing feedback until the component perfectly matches your design and functional requirements, ensuring pixel-perfect fidelity and optimal code quality.