Tools & Workflow

Which AI coding tools you used, how you integrated them
Cursor - asked big brain models like Opus to generate large and complex components and features. Mostly used Composer for debugging, though occasionally Gemini 3.1 helped out.
Claude + Gemini - asked about architecture and high level problem solving.

MCP Usage

Which MCPs you used (if any), what they enabled
None

Effective Prompts

3-5 prompts that worked well (include the actual prompts)

1. Still having the same issues. Suggest different approaches to maybe even redo the line completely.
    - I was struggling massively with debugging line drawing. I asked it to debug several times but asking in this way caused it to reason about the cause of the issue

2. Create a prompt to migrate my architecture to use Firebase RTDB for cursor sync and supabase for auth and board state
    - This created a clear list of instructions to migrate - though I still had to debug some minor issues.

3. Implement frames like miro does
    - It pulled the miro docs on frames and implemented them to a T - even the keyboard shortcut to make one (press F)

Code Analysis Rough

% of AI-generated vs hand-written code
99% AI-generated. I mostly wrote configuration/environment variables.

Strengths & Limitations

Where AI excelled, where it struggled
AI is great at writing logic and implementing libraries. It's not great at understanding the experience of the user - although in some instances it was smart enough to pick up on use

case. For example, I asked to make a change on the login page and it implemented the same design change to the signup page.

Key Learnings

Insights about working with coding agents

It's really easy to accidentally eat up a ton of context. Spin up new chats as much as possible. Use prompts to make prompts - it makes it easy to review and understand what the LLM is expecting to do and have it execute it in a way the LLM understands.