

CollabCanvas — Building Real■Time Collaborative Design Tools with AI

(Challenge 1 — Announcement through Day 3)

Why This Matters

The future of design tools isn't just collaborative — it's **co■creative**. You'll be building the foundation for how humans and AI can design together, in real time.

Project Overview

This is a one■week sprint with three key deadlines:

- **MVP:** Tuesday (24 hours)
- **Early Submission:** Friday (4 days)
- **Final:** Sunday (7 days)

You'll build in two phases: first the **core collaborative canvas** with real■time sync, then an **AI agent** that manipulates the canvas using natural language.

MVP Requirements (24 Hours)

This is a hard gate. To pass the MVP checkpoint, you must have:

- ■ Basic canvas with pan/zoom
- ■ At least one shape type (rectangle, circle, or text)
- ■ Ability to create and move objects
- ■ Real■time sync between 2+ users
- ■ Multiplayer cursors with name labels
- ■ Presence awareness (who's online)
- ■ User authentication (users have accounts/names)
- ■ Deployed and publicly accessible

Focus: collaborative infrastructure.

The MVP isn't about piling on features — it's about proving your foundation is solid. A **simple canvas with bulletproof multiplayer** is worth more than a **feature-rich canvas with broken sync**.

Example Architecture

At minimum, you should have:

- 1 **A backend** (Firestore, Supabase, or custom WebSocket server) that broadcasts updates.
- 2 **A frontend listener** that updates local canvas state and rebroadcasts deltas.
- 3 **A persistence layer** that saves the current state on disconnects.

Core Collaborative Canvas

Canvas Features

Your canvas needs a large workspace with a smooth pan and zoom. It doesn't need to be truly infinite, but it should feel spacious. Support basic shapes — **rectangles, circles, and lines** with solid colors. Add **text layers** with basic formatting.

Users should be able to **transform objects** (move, resize, rotate). Include selection for single and multiple objects (**shift-click** or **drag-to-select**). Add **layer management** and basic operations like **delete** and **duplicate**.

Real-Time Collaboration

Every user should see **multiplayer cursors with names** moving in real time. When someone creates or modifies an object, it appears instantly for everyone. Show clear **presence awareness** of who's currently editing.

Handle **conflict resolution** when multiple users edit simultaneously. (A "last write wins" approach is acceptable, but document your choice.)

Manage **disconnects and reconnects** without breaking the experience. Canvas state **must persist** — if all users leave and come back, their work should still be there.

Testing Scenario

We'll test with:

- 1 ****Two users**** editing simultaneously in different browsers.
- 2 ****One user refreshing mid■edit**** to confirm state persistence.
- 3 ****Multiple shapes**** being created and moved rapidly to test sync performance.

Performance Targets

- Maintain ****60 FPS**** during all interactions (pan, zoom, object manipulation).
- ****Sync object changes**** across users in ****< 100 ms**** and ****cursor positions**** in ****< 50 ms****.
- Support ****500+ simple objects**** without FPS drops and ****5+ concurrent users**** without degradation.

We'll test performance on your ****deployed app****, so make sure it works under load.

AI Canvas Agent

The AI Feature

Build an AI agent that manipulates your canvas through natural language ****using function calling****.

When a user types ****“Create a blue rectangle in the center,”**** the AI agent calls your canvas API functions, and the rectangle appears on everyone's canvas via real■time sync.

Required Capabilities

Your AI agent must support ****at least 6 distinct commands**** showing a range of ****creation****, ****manipulation****, and ****layout**** actions.

Creation Commands (examples)

- “Create a ****red circle**** at position ****100, 200****.”
- “Add a ****text layer**** that says ****‘Hello World’****.”
- “Make a ****200×300 rectangle****.”

Manipulation Commands (examples)

- ****“Move”** the blue rectangle to the ****center****.”

- “**Resize** the circle to be **twice as big**.”
- “**Rotate** the text **45 degrees**.”

Layout Commands (examples)

- “Arrange these shapes in a **horizontal row**.”
- “Create a grid of **3x3 squares**.”
- “**Space** these elements **evenly**.”

Complex Commands (examples)

- “Create a **login form** with **username** and **password** fields.”
- “Build a **navigation bar** with **4 menu items**.”
- “Make a **card layout** with **title, image, and description**.”

Example Evaluation Criteria

When you say “Create a login form,” we expect the AI to create at least **three inputs** (username, password, submit), arranged neatly — not just a text box.

Background (Context)

Figma revolutionized design by making collaboration seamless: multiple designers could work together in real time, **seeing each other’s cursors** and making edits simultaneously **without merge conflicts**.

This required solving complex technical challenges: **real-time synchronization**, **conflict resolution**, and **60 FPS performance** while streaming data across the network.

Now imagine **adding AI** to this. What if you could tell an AI agent to “**create a login form**” and watch it build the components on your canvas? Or say “**arrange these elements in a grid**” and see it happen automatically?

This project challenges you to build both the **collaborative infrastructure** and an **AI agent** that can manipulate the canvas via natural language.

Notes

- Screenshots covered the announcement through ****Day 3****.
- If later pages include “Technical Implementation” or additional constraints, append them below as you receive them.