🎨 Assignment Brief: Simple 2D Canvas Editor

🧠 Overview

We're building tools that simplify work for real users. This challenge is designed to test not just your **coding ability**, but also your **product instincts** and ability to make thoughtful trade-offs.

Your task is to build a **lightweight, web-based canvas editor** where anyone can add and edit shapes/text.

🚀 What You'll Build

✨ Core Concept

A 2d canvas editor built with:

- **React**
- **Fabric.js**
- **Firebase Firestore**

🔧 Core Requirements

1. Home page

- Just a basic landing page, nothing fancy, with a button to create a new canvas.
- On clicking that button, create a new Firestore document, and use that document ID as canvas ID and open the Canvas editor page with url something like /canvas/:canvasId

2. Canvas Editor (Fabric.js)

- A canvas area where users can:
    - Add rectangles, circles, text and a Pen tool
    - Move, resize, rotate, and delete objects
    - Edit text or colours of selected objects

2. Save to Firestore

- Add a save button that saves the canvas to Firestore
- And whenever the same URL is opened, the same canvas is then fetched from Firestore, and we can continue editing. Use the canvasId coming from the URL param to fetch the document

---

🛠️ Tech Stack

- **React**
- **Fabric.js**
- **Firebase Firestore (No Auth Required)**
- Deployment (additional):
    - Firebase Hosting
    - Vercel
    - Any other service of your choice

🧪 Evaluation Criteria

Area	What We're Looking For
Code Quality	Modular components, proper hooks, clean logic
Canvas Logic	Smooth object manipulation, minimal bugs
Persistence	Firebase usage, efficient data saves, readable schema
Routing	URL-based scene IDs, correct handling of /canvas/:id
UX & UI	Intuitive controls, thoughtful design
Product Thinking	Did you go beyond spec to add delightful touches?

📦 Submission Instructions

1. **(REQUIRED)** Share your **GitHub repository** with:
    - A brief **README** explaining what you built and any special mentions
2. **(REQUIRED)** Record a screen recording video explaining your project and showing around what you built. [try to keep it under 5 minutes]
3. (OPTIONAL) Share a **live demo link** (on Firebase Hosting, Vercel, or Netlify)

🧷 Pro Tips

- Think about **how real users would use this** — what would confuse them? What would feel magical?
- Keep your **component architecture clean and extensible**
- Don't over-engineer — the best solutions are simple and thoughtful
- Don't just think from an engineering perspective; have a user-first ideology