

Assignment Brief: Stateless 2D Editor with Shareable Links

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, and **share the canvas via a public link**. No login required — the experience should feel like a “mini Canva” that lives behind a link.

What You'll Build

Core Concept

A stateless, shareable canvas editor built with:

- React
 - Fabric.js
 - Firebase Firestore
-

Core Requirements

1. 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. Scene Management (Stateless URL System)

- When a user opens `/`, generate a new scene ID and redirect to `/canvas/:id`
- Canvas state is saved in **Firestore** under that scene ID. Implement auto-save.
- When `/canvas/:id` is opened, load the corresponding canvas state
- Save all updates back to Firestore in real time (or debounce to reduce writes)

3. Shareability

- Provide a **"Share Canvas"** button
 - Anyone with the link can access and edit the canvas — no login needed
-

Bonus Points (Optional, but appreciated)

Feature	Description
Snap to grid	Align shapes easily with visual snapping
Undo/Redo	Support reversing recent actions
Export	Allow users to export the canvas as PNG or SVG
View-only Mode	Load canvas in read-only mode via <code>?viewOnly=true</code>
Object Locking	Lock/unlock objects to prevent accidental edits
Templates	Let users choose a template when starting a new canvas

Tech Stack

- React
- Fabric.js
- **Firebase Firestore (No Auth Required)**
- Deployment (anyone):
 - 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 <code>/canvas/:id</code>
UX & UI	Intuitive controls, thoughtful design
Product Thinking	Did you go beyond spec to add delightful touches?

Submission Instructions

1. Share your **GitHub repository** with:
 - A brief **README** explaining:
 - What you built
 - What trade-offs did you make?
 - Any bonus features you included
 2. Share a **live demo link** (on Firebase Hosting, Vercel, or Netlify)
 3. Add a Video recording to showcase your solution
-

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**
- Optimise Firestore writes (debounce where needed)
- Don't over-engineer — the best solutions are simple and thoughtful
- Don't just think from an engineering perspective; have a user-first ideology