Build an AI Image Editor with Next.js & Fabric.js :
https://www.youtube.com/watch?v=vr1JL6Fi6K4

Below is the **10-Day "Commando" Sprint** for one full-stack developer.

---

## 🛠️ The "Lethal" Tech Stack (Speed & Stability)

- **Framework:** Next.js 14/15 (App Router)
- **Database/Auth:** Supabase (It's faster to set up than Firebase right now)
- **AI Orchestration:** Google Vertex AI Node.js SDK (`@google-cloud/vertexai`)
  - **Logic:** Gemini Pro 1.5 (or 2.0 Flash if available) for prompt engineering.
  - **Visuals:** Imagen-4 (via Vertex AI Model Garden).
- **Canvas Engine:** `fabric.js` (v6) - *Industry standard for web-based image editors.*
- **Scraping:** `microlink` (API) or `puppeteer` (Self-hosted). *Recommendation: Microlink for Day 1 speed.*

---

## 📅 The 10-Day Execution Plan

### Day 1: The "Hollow" Shell (Infrastructure)

**Goal:** Deploy a working empty app with Auth and Database.

- **Morning (08:00 - 12:00):**
  - Initialize Next.js repo with `shadcn/ui` (Components library).
  - Set up Supabase project (Auth: Google/Email).
  - **Crucial:** Create the Database Schema: `profiles` (stores brand info), `projects` (stores canvas JSON), `generations` (logs AI usage).
- **Afternoon (13:00 - 17:00):**
  - Build the "Higgsfield" Landing Page skeleton.
  - Implement the "Magic Input" (The big URL bar).
  - **Deliverable:** A site live on Vercel where you can log in.

### Day 2: The "Brand DNA" Extraction (The Magic)

**Goal:** User enters URL -> We get Hex Codes, Fonts, and Logo.

- **Morning:**
  - Build API Route `/api/extract-brand`.

    - **Tool:** Use `cheerio` to scrape the HTML. Look for `og:image` (Logo), and scan CSS files for `font-family`.
    - **Fallback:** If scraping fails, default to a standard "Tech" palette.
- **Afternoon:**
    - Build the "Brand Confirmation" Modal.
    - *Dev Note:* Allow users to upload a logo if the scraper grabs the wrong one.
    - **Deliverable:** Enter "apple.com" -> System returns Grey/Black/White and the Apple logo.

**Day 3: The Canvas Engine (Fabric.js Setup)**

**Goal:** A white box where we can drag/drop things.

- **Morning:**
    - Initialize `fabric.Canvas` in a React `useEffect` hook.
    - Implement "Responsive Canvas" (It must resize to fit the screen but keep 1080x1350 aspect ratio).
- **Afternoon:**
    - Create the "Template Logic."
    - **Hard Requirement:** Define the 4 templates as JSON coordinates.
        - *Template 1:* Image fills canvas.
        - *Template 2:* Image 100%, Text Block at `top: 100`, `left: 50`.
    - **Deliverable:** A page with a white canvas where you can manually add text.

**Day 4: The Brain (Gemini System Prompt)**

**Goal:** Converting "I want a post about coffee" into a lethal Imagen prompt.

- **Morning:**
    - Set up Vertex AI SDK.
    - Write the **Prompt Chain** (See specific prompt below).
    - *Constraint:* The prompt must instruct Imagen to "leave negative space" for text if using Templates 2-4.
- **Afternoon:**
    - Connect the "Style Chips" (Example prompts).
    - **Deliverable:** You type "Coffee", and the console logs: *"Photorealistic close-up of espresso, steam rising, dark moody lighting, negative space in top center..."*

**Day 5: Connecting Imagen-4**

**Goal:** Generating the pixels.

- **All Day:**
    - Connect the `generateImage` endpoint.

    - **Critical Dev Task:** Handle the "Base64" response. Store the image in Supabase Storage buckets immediately (don't rely on the temporary URL).
    - Display the 4 variants in a grid.
    - **Deliverable:** You type a prompt -> 4 images appear.

**Day 6: The "Merge" (Image + Canvas)**

**Goal:** Clicking an image puts it into the Editor.

- **Morning:**
    - On "Select Variant": Load the Imagen result as the *Background Image* of the Fabric Canvas.
- **Afternoon:**
    - **The "Smart Overlay":** Automatically inject the Headlines (generated by Gemini) onto the canvas using the **User's Brand Font**.
    - *Why this is seamless:* The user sees the image *and* the text instantly. They can click the text to fix a typo.
    - **Deliverable:** The "MVP" Loop is complete. URL -> Brand -> Prompt -> Image + Text on Screen.

**Day 7: The "Imagen Text" Mode (Special Feature)**

**Goal:** Enabling the "Text Baking" for users who want cool AI text effects.

- **Morning:**
    - Add a toggle in UI: "Use AI Text Effects?" (Warning: Non-Editable).
    - Modify System Prompt: If checked, instruct Imagen-4 to render the text: *"Text 'SALE' written in smoke"*.
- **Afternoon:**
    - Test this rigorously. Imagen-4 is great, but it requires specific prompting ("Render the word: 'XYZ'").

**Day 8: Editor Refinement (The "Feel")**

**Goal:** Making it feel like a pro tool, not a dev project.

- **Morning:**
    - Add "Toolbar": Change Font Color, Size, Font Family.
    - Add "Regenerate Text": A button that asks Gemini to rewrite the headline *without* changing the image.
- **Afternoon:**
    - Implement "Download": Export Canvas to PNG/JPG (High Res).

**Day 9: Polish & Dashboard**

**Goal:** Saving projects.

- **All Day:**
    - "My Projects" page: Fetch saved JSON states from Supabase.
    - "Remix": Button to load an old project state into the editor.

**Day 10: Bug Hunt & Launch**

**Goal:** Nothing breaks.

- **Morning:**
    - Stress test: What happens if the URL scrape fails? (Add error toasts).
    - What happens if Imagen API times out? (Add retry logic).
- **Afternoon:**
    - Deploy.

---

## 🧠 The "Secret Sauce" System Prompt (For Day 4)

*Give this exact prompt structure to your developer to use with Gemini Pro.*

Plaintext

```
Role: You are a world-class Visual Designer and Prompt Engineer for Imagen-4.

Context:
- User Brand Colors: [List of Hex Codes]
- User Brand Vibe: [Extracted from site, e.g., "Minimalist Tech"]
- User Request: "[User Input]"
- Template Mode: [Headline / No Text]

Task:
1. Create a detailed image generation prompt for Imagen-4.
2. If Template Mode requires text, ensure the prompt explicitly asks for "clean, empty negative
space" in the appropriate area (top/bottom/side) so we can overlay text later.
3. If the user asked for a specific object, style it to match the Brand Vibe.

Output Format (JSON):
{
  "imagen_prompt": "...",
  "headline_suggestion": "...", // Short, punchy copy for the overlay
  "subheadline_suggestion": "..."
}
```

## 📹 Relevant Resource for the Developer

Since your developer is using Next.js and Fabric.js, this specific tutorial is the closest "blueprint" to what you are building. It covers the "Canvas" setup perfectly.