You're almost there — sign up to start building in Notion today.    **Sign up or login**

# Take-Home Engineering Challenge: Image Transformation Service

**Objective:**

Develop a full-stack application that allows users to upload an image, process it through background removal and horizontal flipping, and then manage the resulting images.

**Core Functionality Requirements:**

1. **Image Upload:**

   - The frontend should provide a clear and intuitive interface for a user to upload a single image file.

2. **Image Processing:**

   - Upon upload, the backend should process the image.

   - **Background Removal:** Integrate with a third-party service to remove the background of the uploaded image. *Please utilize free trials or free credits provided by such services to avoid incurring costs.*

   - **Horizontal Flip:** After background removal, the image should be horizontally flipped.

3. **Image Hosting & Retrieval:**

   - The processed image (background removed and flipped) must be hosted online.

   - The user should receive a unique URL to access their edited image.

4. **Image Deletion:**

   - Provide functionality for the user to delete their uploaded and processed images from the hosting location.

**Technical Requirements:**

- **Backend:** Must be built using **TypeScript**.

- **Deployment:** The completed application (both frontend and backend) must be **deployed online** andaccessible for review. This means providing a live URL.

- **Code Submission:** All source code (frontend and backend) must be shared via a **GitHub repository**.

Most important: Have fun 🙂

**What we are looking at:**

### User Experience & Design:

We want to see how you build a complete product end-to-end. It's not about how much time you invest, but how you prioritize quality. We are looking for an intuitive interface that provides clear feedback (like loading states) and feels like a polished tool ready for a user.

### Backend Engineering:

Using TypeScript, show us your ability to build a robust and clean service. We will focus on:

- **API Structure:** Clean, logical endpoint design and proper error handling.

- **Process Management:** Efficiently handling the flow between the third-party background removal, the horizontal flip, and cloud storage.

- **Code Quality:** Use of types/interfaces, modularity, and secure management of API keys and file uploads.

Example picture from input to expected result: