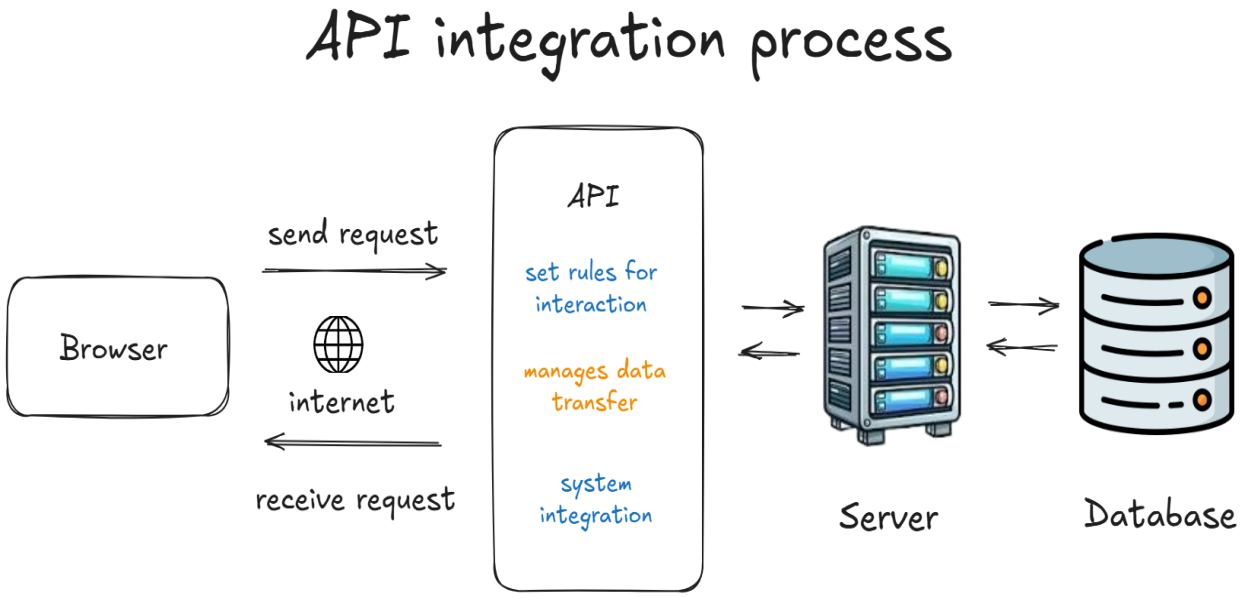
Task for Day 3

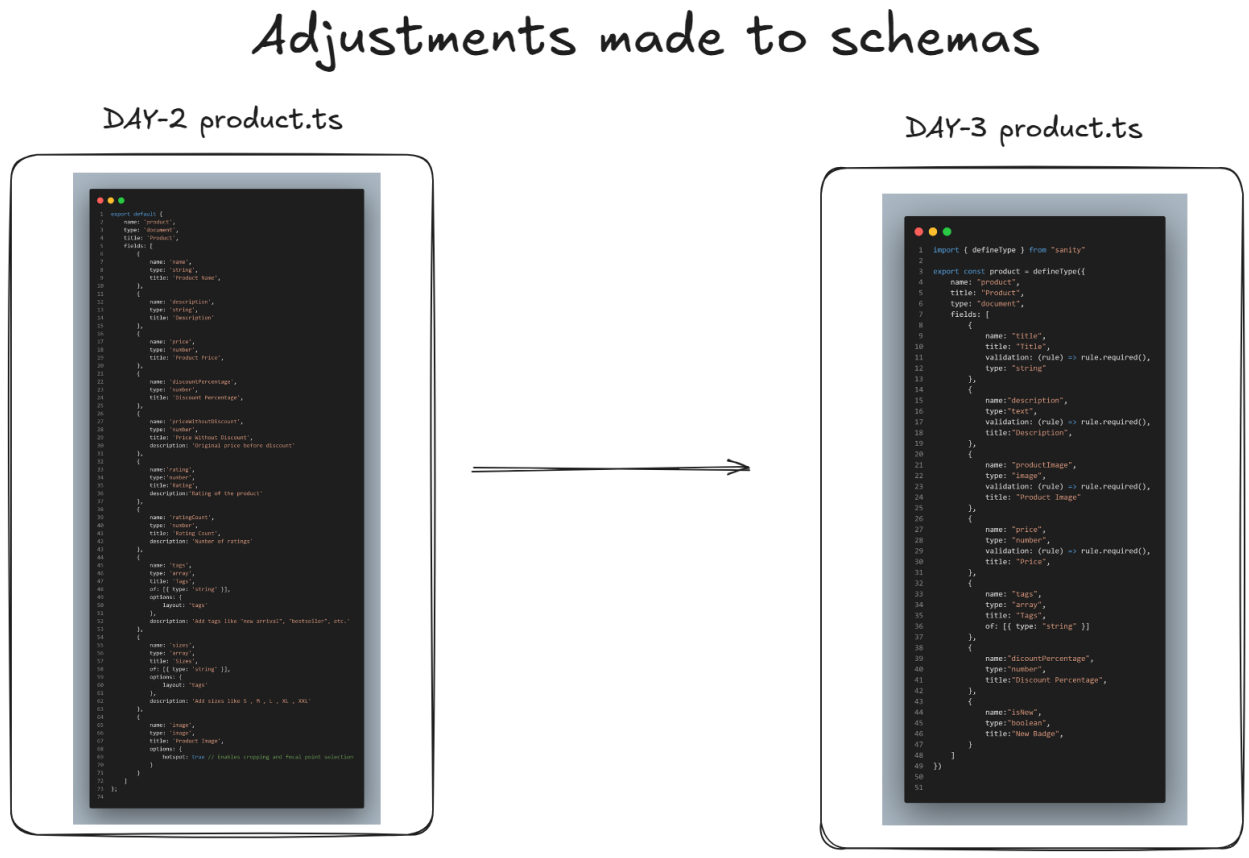
API Integration Report

FURNIRO

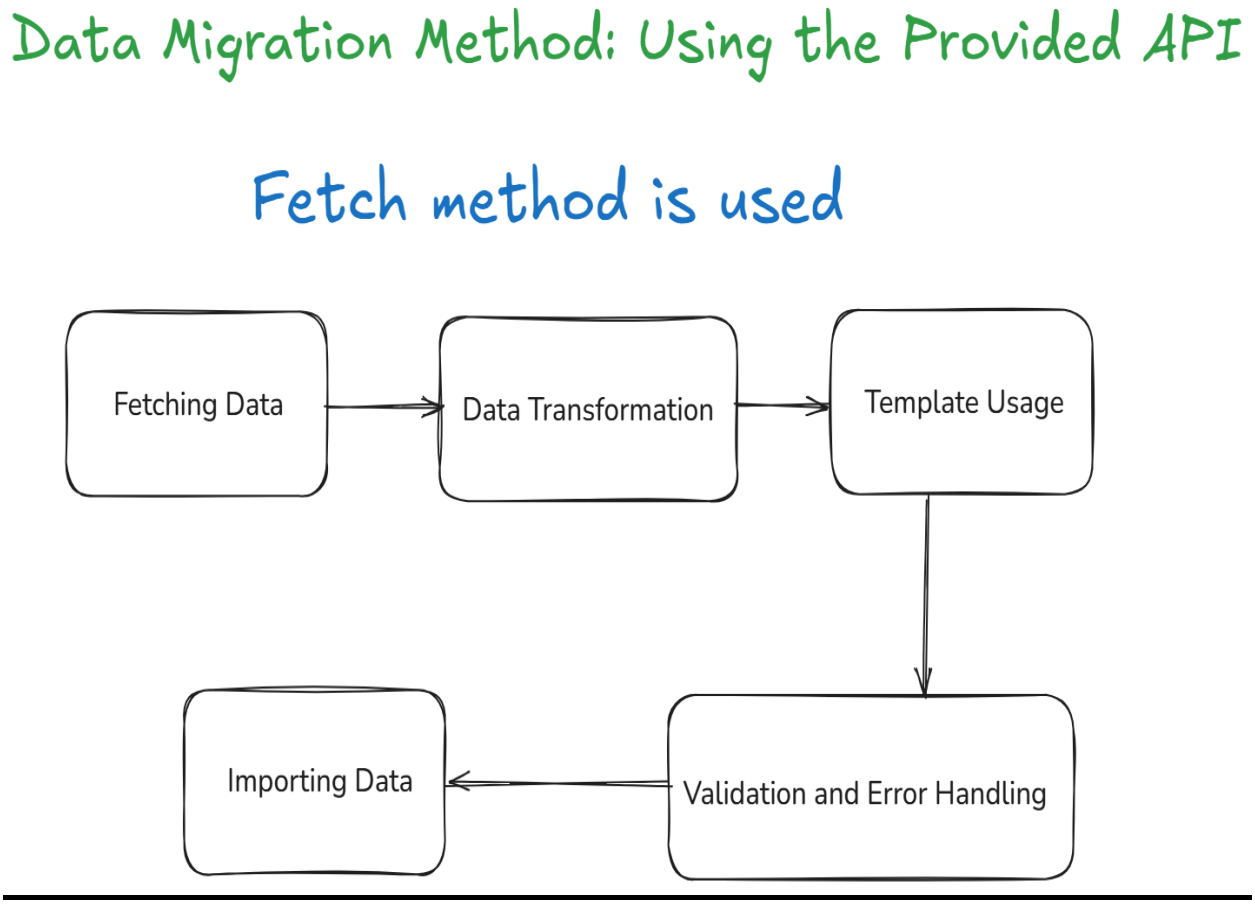
API Understanding



Schemas Adjustments



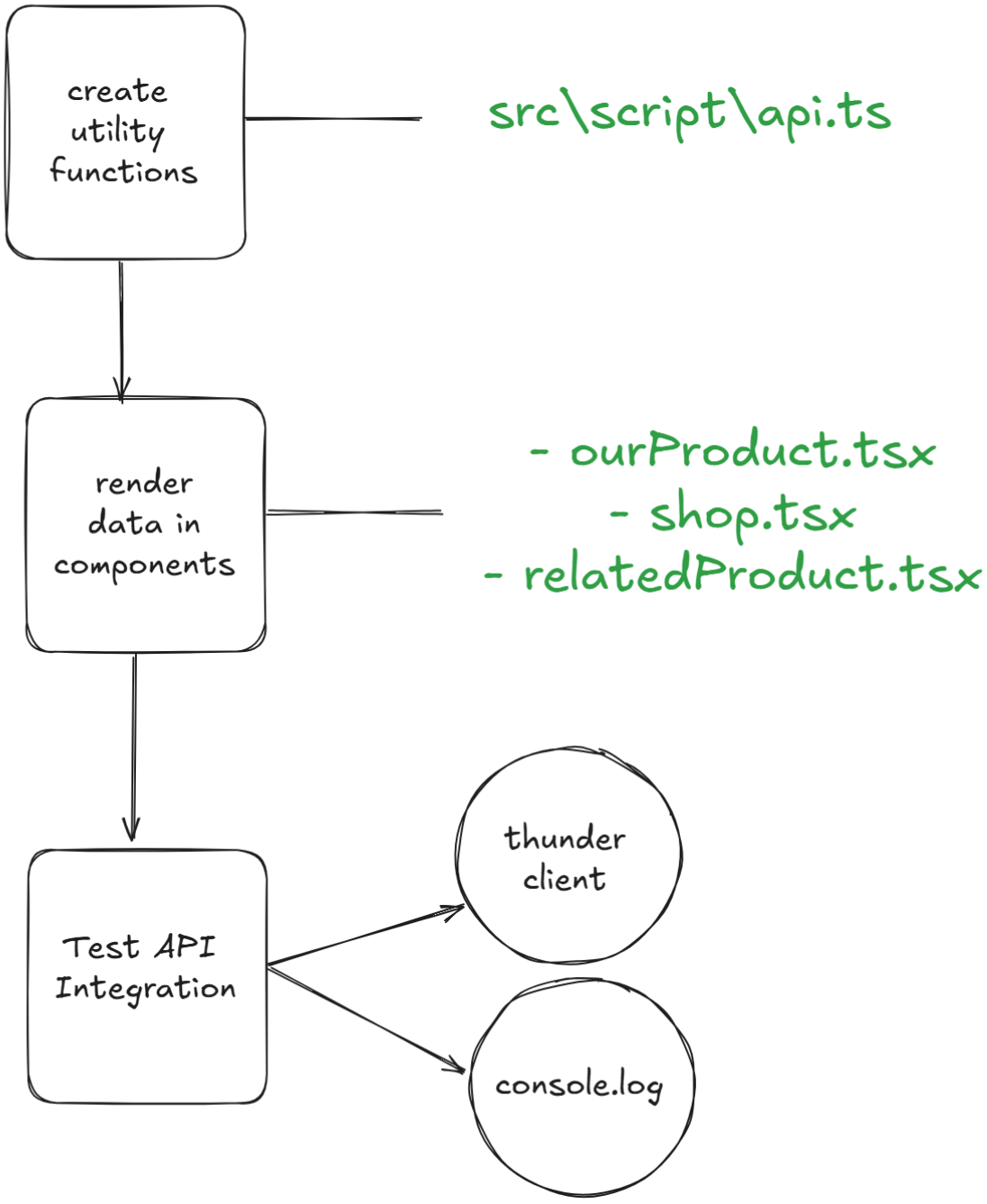
Migration steps



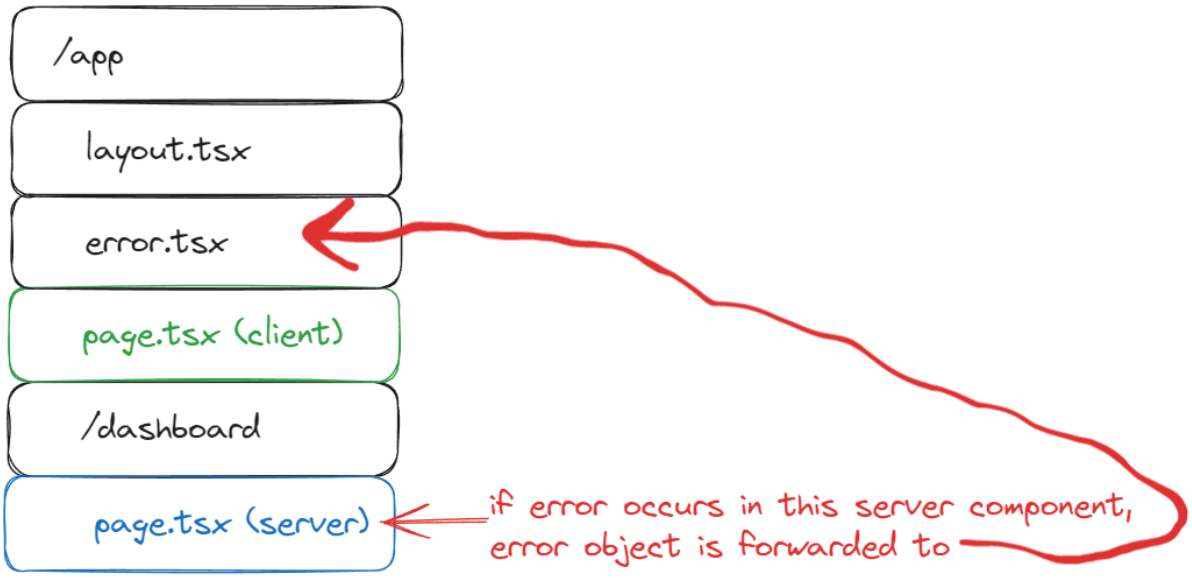
api.ts

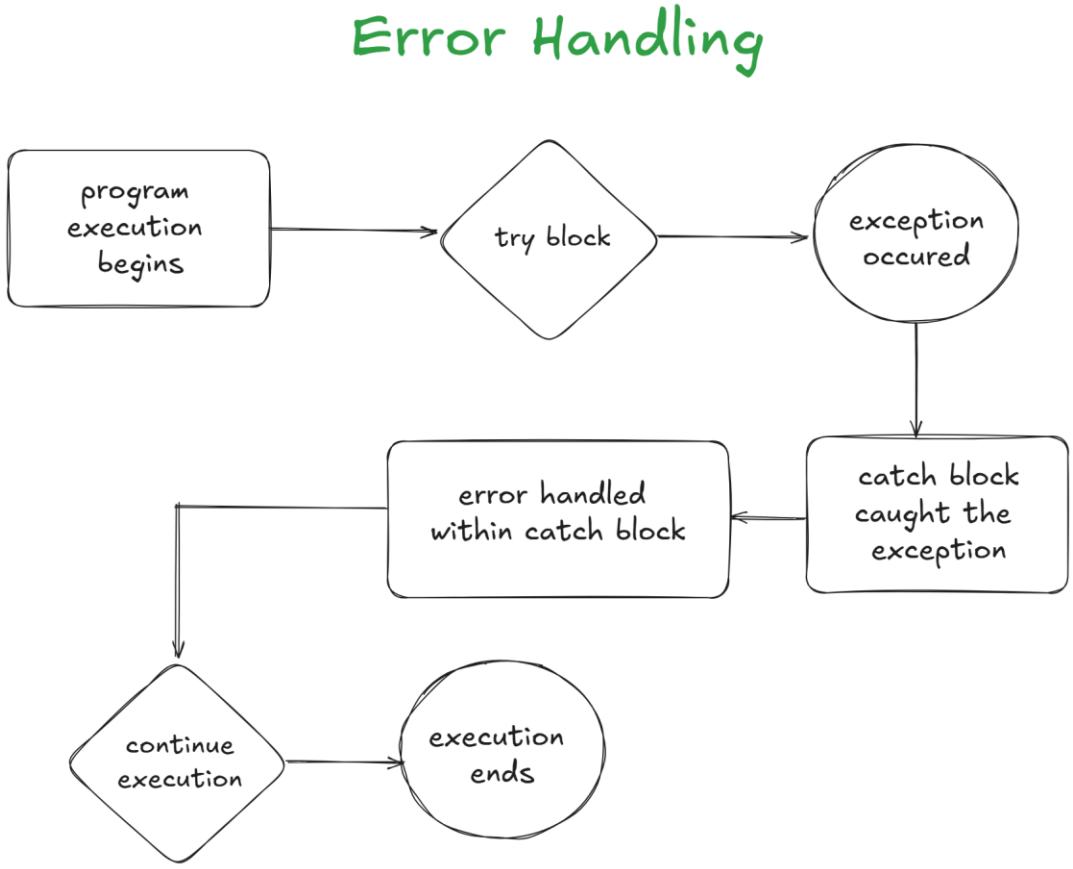
```ts
"use server";

import { client } from "@/sanity/lib/client";

async function uploadImageToSanity(imageUrl: string) {
  const response = await fetch(imageUrl);
  const blob = await response.blob();

  const asset = await client.assets.upload("image", blob);
  return asset;
}

async function saveProductToSanity(product: any, imageAssetId: string) {
  const sanityProduct = {
    _id: `product-${product.id}`,
    _type: "product",
    name: product.title,
    price: product.price,
    description: product.description,
    rating: product.rating?.rate || 0,
    ratingCount: product.rating?.count || 0,
    discountPercentage: product.discountPercentage || 0,
    tags: product.category ? [product.category] : [],
    image: {
      _type: "image",
      asset: {
        _type: "reference",
        _ref: imageAssetId,
      },
    },
  };

  await client.createOrReplace(sanityProduct);
  console.log(`✅ Imported product: ${sanityProduct.name}`);
}

export async function fetchData() {
  // Check if data already exists in localStorage
  const localData = localStorage.getItem("products");

  if (localData) {
    console.log("📦 Using cached data from localStorage");
    return JSON.parse(localData); // Return the cached data
  }

  console.log("🔵 Fetching data from API...");
  const response = await fetch("https://template-6-api.vercel.app/api/products");
  const products = await response.json();

  // Save fetched data to localStorage
  localStorage.setItem("products", JSON.stringify(products));
  console.log("💾 Data saved to localStorage!");

  const promises = products.map(async (product: any) => {
    try {
      const existingProduct = await client.fetch(
        `*[_type == "product" && _id == $id][0]`,
        { id: `product-${product.id}` }
      );

      if (existingProduct) {
        console.log(`🔵 Product already exists: ${product.title}`);
        return;
      }

      const imageAsset = await uploadImageToSanity(product.image);
      await saveProductToSanity(product, imageAsset._id);
    } catch (error) {
      console.error(`❌ Failed to process product: ${product.title}`, error);
    }
  });

  await Promise.all(promises);

  console.log("✅ All products imported successfully!");

  return products; // Return the fetched data
}
```

API Integration in Next.js



Error Handling





Checklist for day3

- API-understanding ✓
- Schema Validation ✓
- Data Migration ✓
- API Integration in Next.js: ✓
- Submission Preparation ✓