Day 3 - API Integration Report - Regal House

1. API Integration Process:

- The third-party API used for this project is hosted by https://template-0-beta.vercel.app/api/product. This API provides product-related data, such as the name, description, price, image path, discount, category, stock level, and whether the product is featured. This API is integrated into sanity to populate data.
- To display images correctly from Sanity, a remote pattern was added to the "next.config.mjs" file to allow Next.js to fetch external images.

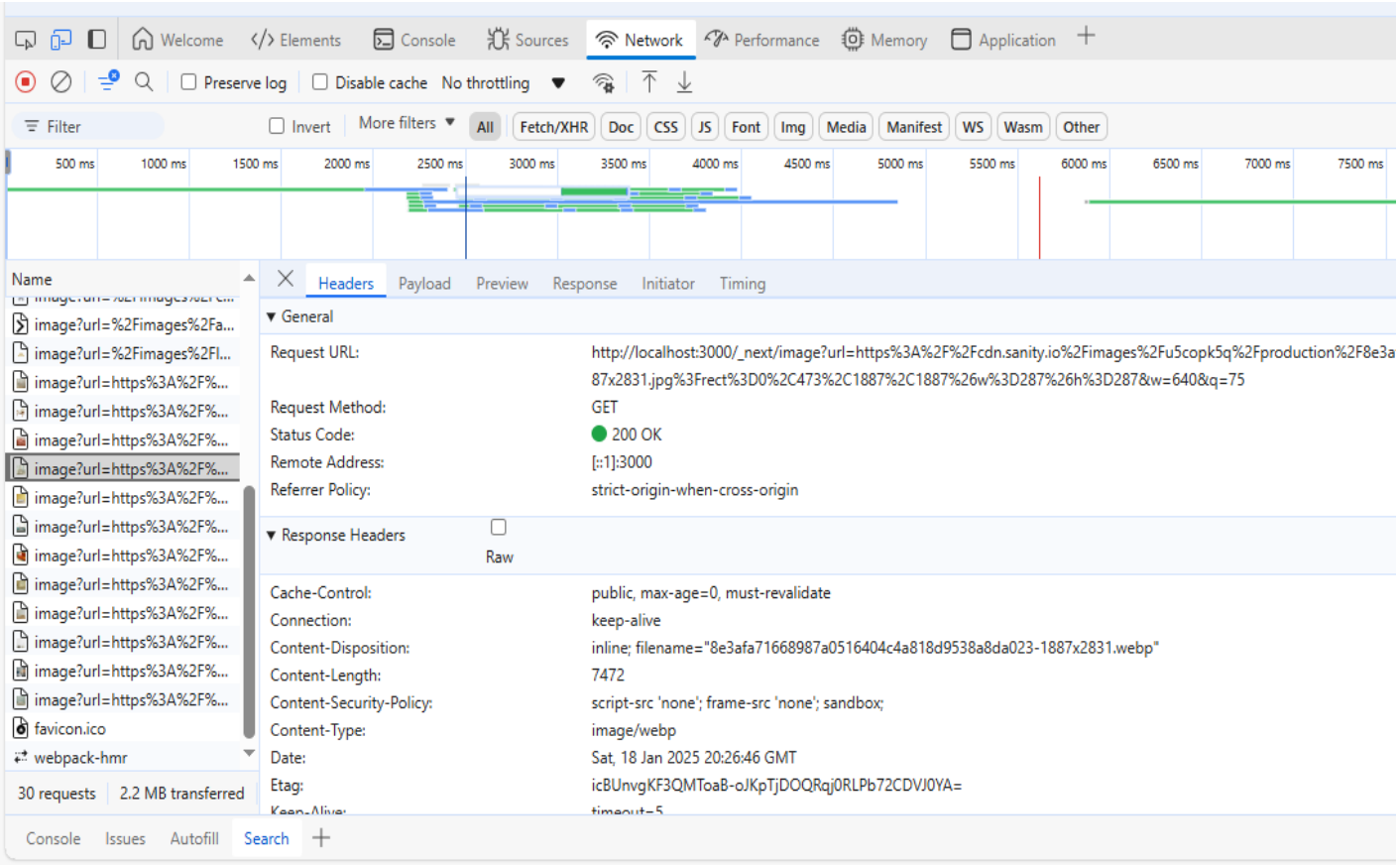
2. Adjustments Made to Sanity Schema:

- Initially, the sanity product schema was designed to include basic fields such as name, description and price. However, after integrating the third-party API, the schema was updated to accommodate the additional fields provided by the API.
- However, additional changes will be made to the schema as needed when adding further functionality.

3. Data Migration:

- The data from the third-party API is fetched and then imported into the Sanity CMS. A script is used to map the API data to the corresponding schema fields in Sanity.
- On the frontend, products are fetched using the Sanity client and displayed dynamically.

API Calls


Welcome
Elements
Console
Sources
Network
Performance
Memory
Application
Preserve log
Disable cache
No throttling
Filter
Invert
More filters
All
Fetch/XHR
Doc
CSS
JS
Font
Img
Media
Manifest
WS
Wasm
Other
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
4000 ms
4500 ms
5000 ms
5500 ms
6000 ms
6500 ms
7000 ms
7500 ms
Name
image?url=%2Fimages%2Fa...
image?url=%2Fimages%2Fl...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
image?url=https%3A%2F%...
favicon.ico
webpack-hmr
30 requests
2.2 MB transferred
Headers
Payload
Preview
Response
Initiator
Timing
General
Request URL:
http://localhost:3000/_next/image?url=https%3A%2F%2Fcdn.sanity.io%2Fimages%2Fu5copk5q%2Fproduction%2F8e3a
87x2831.jpg%3Frect%3D0%2C473%2C1887%2C1887%26w%3D287%26h%3D287&w=640&q=75
Request Method:
GET
Status Code:
200 OK
Remote Address:
[::1]:3000
Referrer Policy:
strict-origin-when-cross-origin
Response Headers
Raw
Cache-Control:
public, max-age=0, must-revalidate
Connection:
keep-alive
Content-Disposition:
inline; filename="8e3afa71668987a0516404c4a818d9538a8da023-1887x2831.webp"
Content-Length:
7472
Content-Security-Policy:
script-src 'none'; frame-src 'none'; sandbox;
Content-Type:
image/webp
Date:
Sat, 18 Jan 2025 20:26:46 GMT
Etag:
icBUnvgKF3QMToaB-oJKpTjDOQRqj0RLPb72CDVJ0YA=
Keep-Alive:
timeout=5
Console
Issues
Autofill
Search

Data Displayed on the Frontend

Image 1:

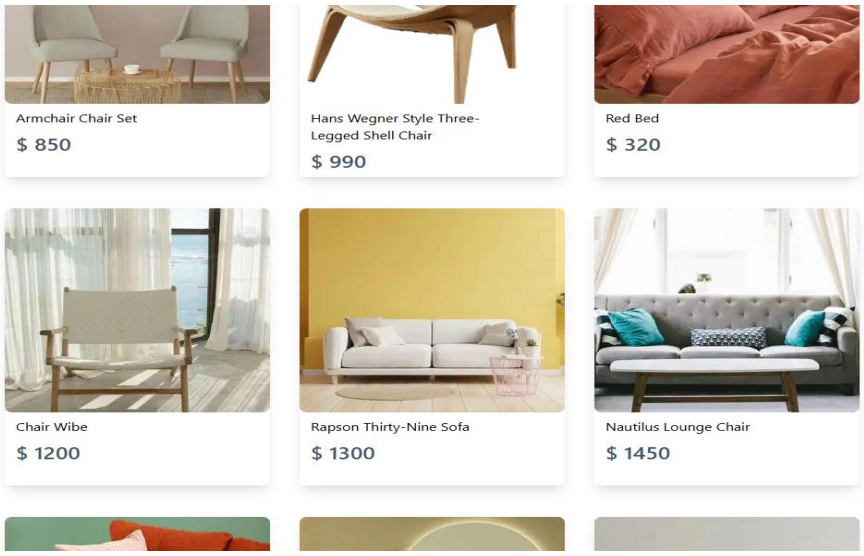


Image 2:

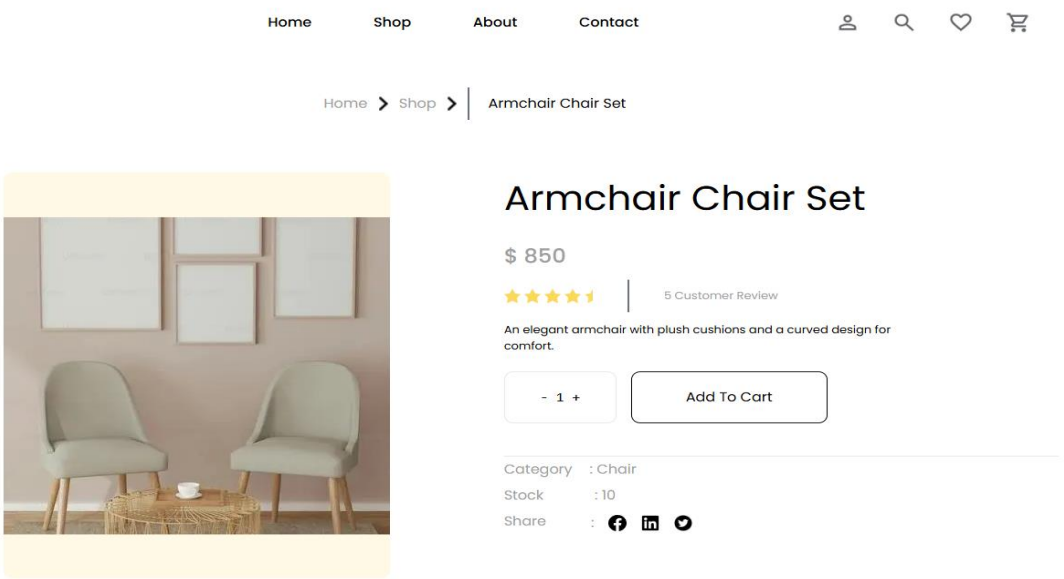


Image 3:

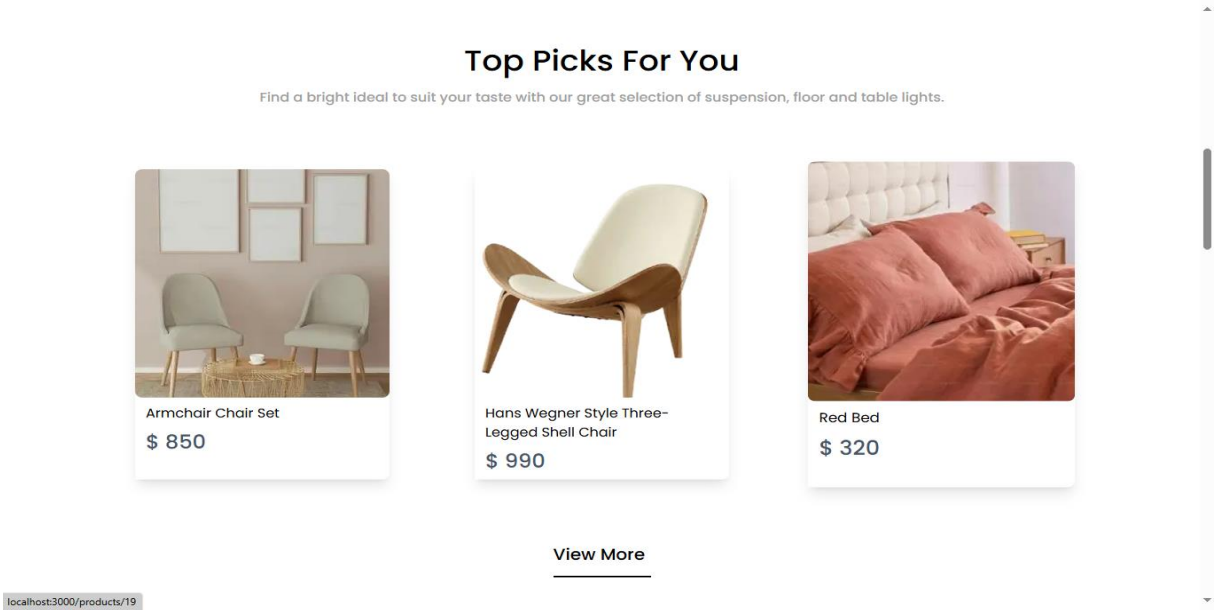
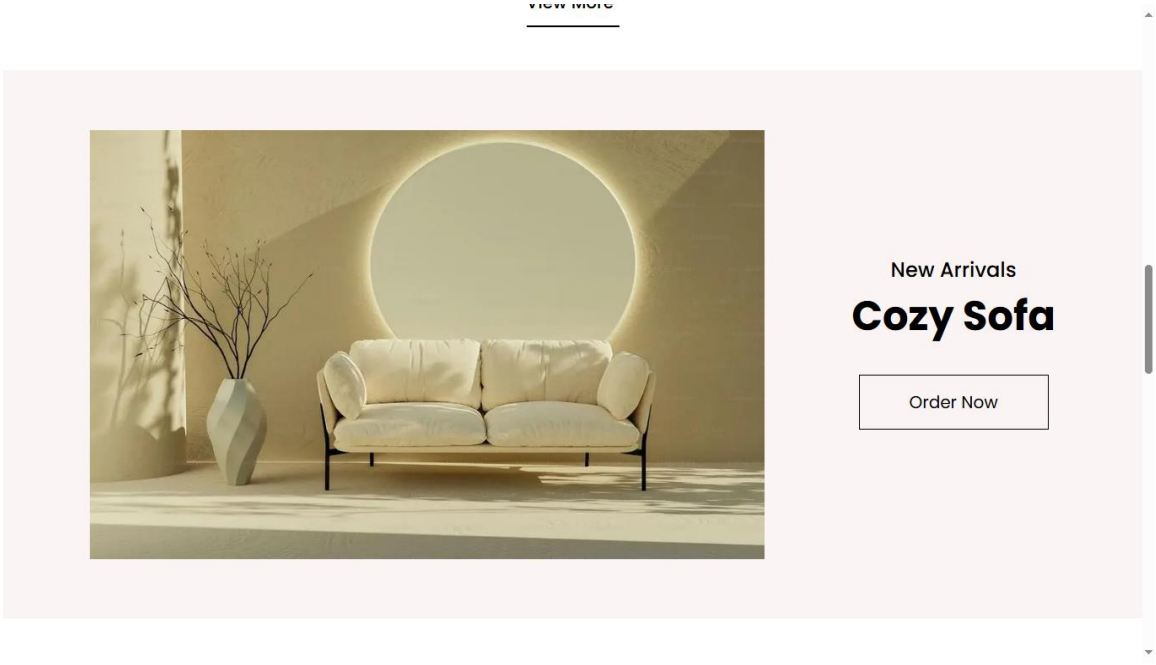
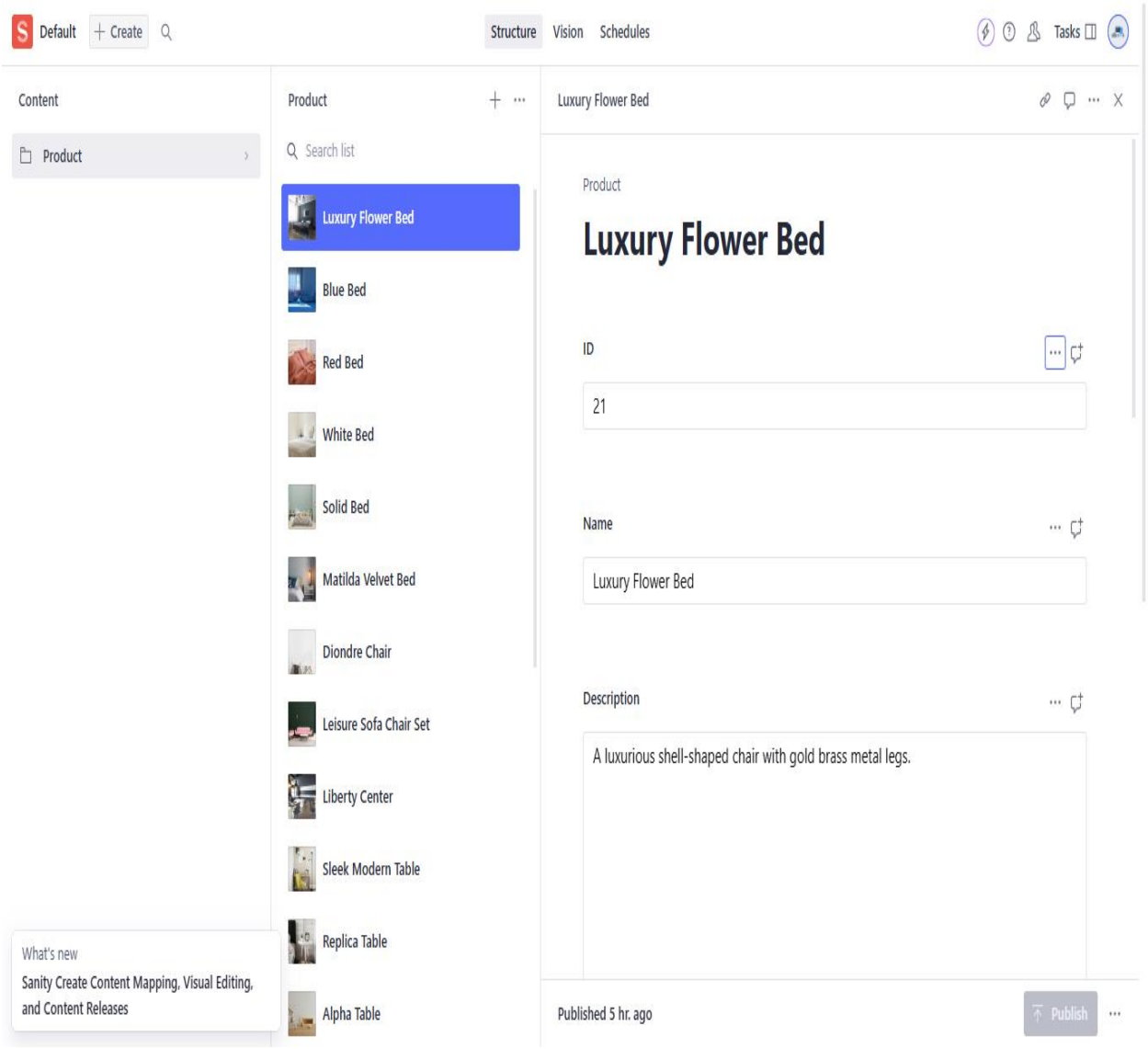


Image 4:



Populated Sanity CMS Fields



Code snippets for migration scripts.

```
async function uploadImageToSanity(imagePath) {
  try {
    console.log(`Uploading image: ${imagePath}`);

    const response = await fetch(imagePath);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imagePath}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload("image", bufferImage, {
      filename: imagePath.split("/").pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error("Failed to upload image:", imagePath, error);
    return null;
  }
}

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imagePath);

    if (imageId) {
      const document = {
        _type: "product",
        id: product.id,
        name: product.name,
        price: product.price,
        imagePath: {
          _type: "image",
          asset: {
            _ref: imageId,
          },
        },
```

```
        discountPercentage: product.discountPercentage,
        description: product.description,
        category: product.category,
        stockLevel: product.stockLevel,
      };

      const createdProduct = await client.create(document);
      console.log(
        `Product ${product.name} uploaded successfully:`,
        createdProduct
      );
    } else {
      console.log(
        `Product ${product.name} skipped due to image upload failure.`
      );
    }
  } catch (error) {
    console.error("Error uploading product:", error);
  }
}
```

Code snippets for API Integration.

```
async function importProducts() {
  try {
    const response = await fetch(
      "https://template-0-beta.vercel.app/api/product"
    );

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error("Error fetching products:", error);
  }
}
```

```
importProducts();
```