Day 3 - API Integration Report - Furniro

Table of Contents

Setting up environment variable

```
NEXT_PUBLIC_SANITY_PROJECT_ID=your_project_id
NEXT_PUBLIC_SANITY_DATASET=production
SANITY_API_TOKEN=your_sanity_token

Ensures .env.local file contains all of these generate token from sanity

Steps http://localhost:3000/studio -> click on profile icon on top right then
find API click on it and then click on token generate it select developer
while selecting role.
```

Scripts to migrate API data to sanity

First create a scripts folder inside folder create file importSanityData.mjs file write script code there ensures to adjust script code according to API or according to data importSanityData.mjs code will be last of the document.

Then after customization code on package.json file write this on

```
"import-data": "node scripts/importSanityData.mjs"
```

Now package.json looks like

```
"scripts": {
  "dev": "next dev",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "import-data": "node scripts/importSanityData.mjs"
},
```

Now run `npm run import-data` after this command data will start migrating from api to sanity. Sanity we write qrop query to render data on frontened

Code snippets for API integration and migration scripts.

Migration from API to Sanity

```javascript
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

// Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../.env.local') });

// Create Sanity client
const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: false,
  token: process.env.SANITY_API_TOKEN,
  apiVersion: '2021-08-31',
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}
```

```javascript
async function importData() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://template6-
six.vercel.app/api/products');
    const products = response.data;
    console.log(`Fetched ${products.length} products`);

    for (const product of products) {
      console.log(`Processing product: ${product.title}`);

      // Upload image if available
      let imageRef = null;
      if (product.imageUrl) {
        imageRef = await uploadImageToSanity(product.imageUrl);
      }

      // Map product data to Sanity schema
      const sanityProduct = {
        _type: 'product',
        title: product.title,
        description: product.description,
        price: product.price,
        discountPercentage: product.dicountPercentage || 0,
        isNew: product.isNew || false,
        tags: product.tags || [],
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
        productId: product._id || null,
      };

      console.log('Uploading product to Sanity:', sanityProduct.title);
      const result = await client.create(sanityProduct);
      console.log(`Product uploaded successfully: ${result._id}`);
    }

    console.log('Data import completed successfully!');
```

```
  } catch (error) {
    console.error('Error importing data:', error);
  }
}

// Start the import process
importData();
```

Sanity Schema

```
export default {
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      {
        name: 'title',
        title: 'Title',
        type: 'string',
        description: 'The title of the product',
      },
      {
        name: 'description',
        title: 'Description',
        type: 'text',
        description: 'Detailed description of the product',
      },
      {
        name: 'price',
        title: 'Price',
        type: 'number',
        description: 'The price of the product',
      },
      {
        name: 'discountPercentage',
        title: 'Discount Percentage',
        type: 'number',
        description: 'Discount percentage for the product',
      },
      {
        name: 'tags',
        title: 'Tags',
```

```
      type: 'array',
      of: [{ type: 'string' }],
      description: 'Tags associated with the product',
    },
    {
      name: 'isNew',
      title: 'Is New',
      type: 'boolean',
      description: 'Indicates if the product is new',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: {
        hotspot: true, // Enables image cropping and positioning
      },
      description: 'Main image of the product',
    },
    {
      name: 'productId',
      title: 'Product ID',
      type: 'string',
      description: 'Custom unique identifier for the product',
    },
  ],
};
```

ScreenShots