Day 3 - API Integration Report - SHOP.CO

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`

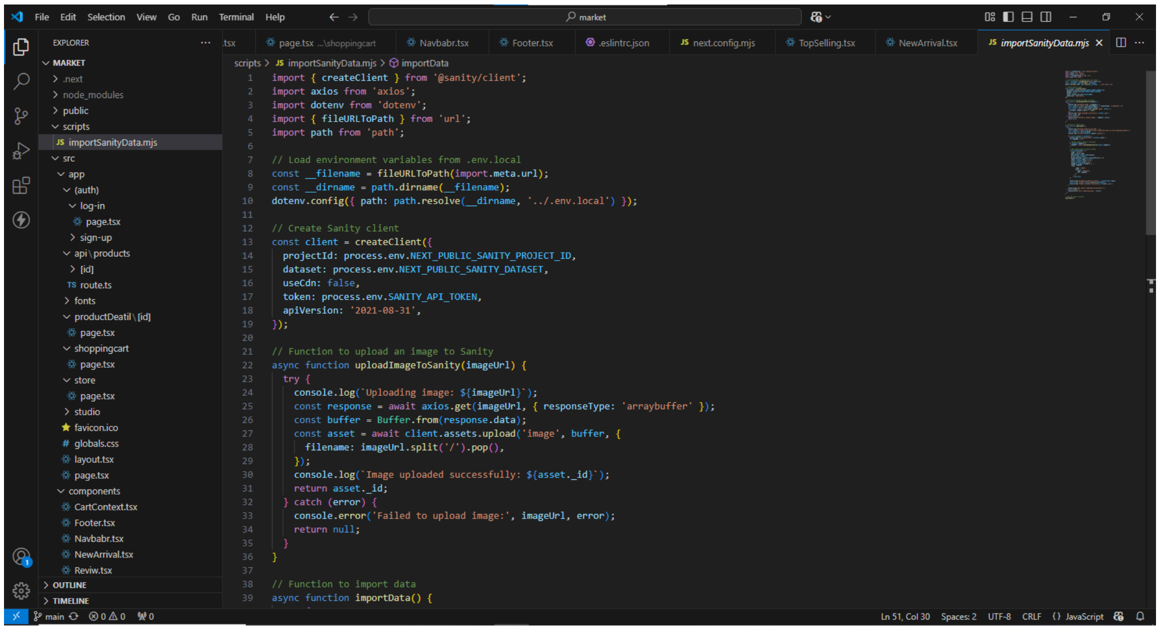
Code snippets for API integration and migration scripts.

Migration from API to Sanity

```
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',
});

// Function to upload an image to Sanity
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;
  }
}

// Function to import data
async function importData() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://template1-neon-
nu.vercel.app/api/products');
    const products = response.data;
    console.log(`Fetched ${products.length} products`);

    for (const product of products) {
      console.log(`Processing product: ${product.name}`);
      let imageRef = null;

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

      // Map product data to Sanity schema
      const sanityProduct = {
        _type: 'product',
        name: product.name,
        description: product.description,
        price: product.price,
        discountPercent: product.discountPercent || 0,
        isNew: product.isNew || false,
        colors: product.colors || [],
        sizes: product.sizes || [],
        category: product.category || '',
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
      };
      console.log('Uploading product to Sanity:', sanityProduct.name);
      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);
  }
}

// Run the import function
importData();
```

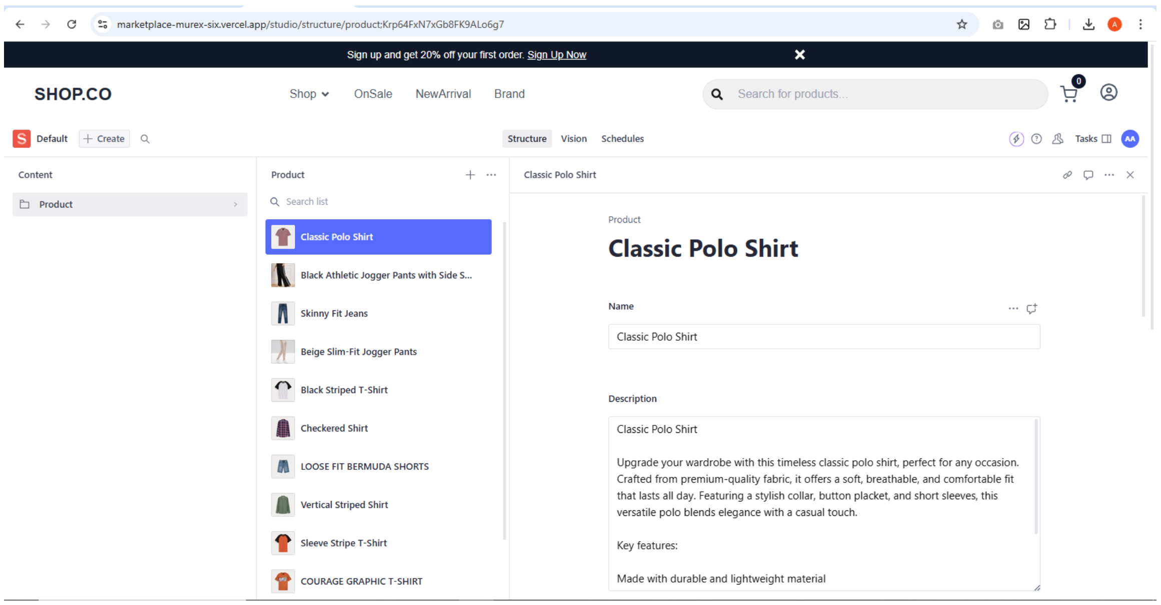
Sanity Schema

```
// schemas/product.js
 const customerSheme={
    name: 'product',
    title: 'Product',
    type: 'document',
    fields: [
      {
        name: 'name',
        title: 'Name',
        type: 'string',
      },
      {
        name: 'description',
        title: 'Description',
        type: 'text',
      },
      {
        name: 'price',
        title: 'Price',
        type: 'number',
      },
      {
        name: 'discountPercent',
        title: 'Discount Percent',
        type: 'number',
      },
      {
        name: 'isNew',
        title: 'Is New',
        type: 'boolean',
      },
      {
        name: 'colors',
        title: 'Colors',
        type: 'array',
        of: [{ type: 'string' }],
      },
      {
        name: 'sizes',
        title: 'Sizes',
        type: 'array',
        of: [{ type: 'string' }],
      },
      {
        name: 'category',
        title: 'Category',
        type: 'string',
      },
      {
        name: 'image',
        title: 'Image',
        type: 'image', // This should match the data structure you're sending
        options: {
          hotspot: true, // Optional, for cropping
        },
      },
    ],
  };
export default customerSheme
```



```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',
});

// Function to upload an image to Sanity
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;
  }
}

// Function to import data
async function importData() {
```

ScreenShots


marketplace-murex-six.vercel.app/studio/structure/product;Krp64FxN7xGb8FK9ALo6g7

Sign up and get 20% off your first order. Sign Up Now

SHOP.CO — Shop · OnSale · NewArrival · Brand — Search for products...

Default · Create — Structure · Vision · Schedules — Tasks

Content
- Product

Product
- Search list
- Classic Polo Shirt
- Black Athletic Jogger Pants with Side S...
- Skinny Fit Jeans
- Beige Slim-Fit Jogger Pants
- Black Striped T-Shirt
- Checkered Shirt
- LOOSE FIT BERMUDA SHORTS
- Vertical Striped Shirt
- Sleeve Stripe T-Shirt
- COURAGE GRAPHIC T-SHIRT

Classic Polo Shirt

Product

Classic Polo Shirt

Name: Classic Polo Shirt

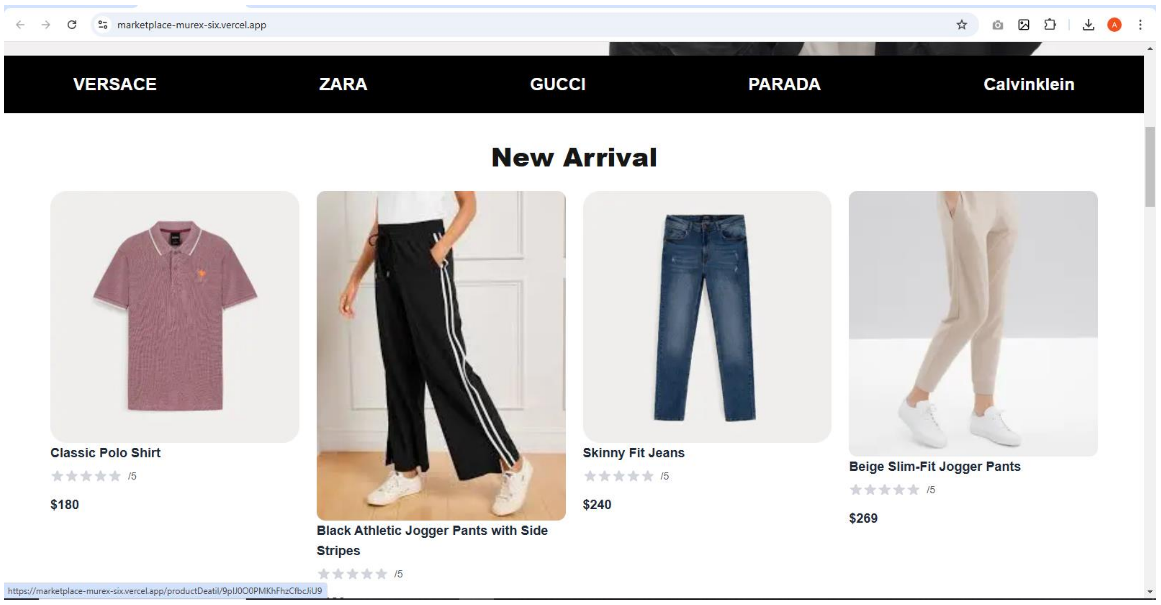
Description:

Classic Polo Shirt

Upgrade your wardrobe with this timeless classic polo shirt, perfect for any occasion. Crafted from premium-quality fabric, it offers a soft, breathable, and comfortable fit that lasts all day. Featuring a stylish collar, button placket, and short sleeves, this versatile polo blends elegance with a casual touch.

Key features:

Made with durable and lightweight material


marketplace-murex-six.vercel.app
VERSACE
ZARA
GUCCI
PARADA
Calvinklein
New Arrival
Classic Polo Shirt
/5
$180
Black Athletic Jogger Pants with Side Stripes
/5
Skinny Fit Jeans
/5
$240
Beige Slim-Fit Jogger Pants
/5
$269
https://marketplace-murex-six.vercel.app/productDeatil/9pIJ0O0PMKhFhzCfbcJiU9