Day 3 - API Integration Report - ShoeVibe

API Integration Process:

1. **Sanity Backend Setup:**
   - Created and configured a Sanity project to serve as the backend for the application.
   - Defined the required schemas (e.g., products, categories) in the Sanity studio for structured data storage.

2. **Next.js API Routes:**
   - Created and configured a Sanity project to serve as the backend for the application.
   - Defined the required schemas (e.g., products, categories) in the Sanity studio for structured data storage.
3. **Fetching Data in the Frontend:**
   - Utilized fetch to make HTTP requests to the Next.js API routes from the frontend.
   - Processed and displayed the data dynamically in the application.

Adjustments made to schemas:

Product Schema:

- Added fields such as id, name, description, price, category, and images.
- Used validation rules to ensure data consistency.

Migration Steps:

Data Preparation:

- Prepared mock product data in JSON format for migration.

Data Migration Script:

- Wrote a migration script in Next.js to push data into Sanity using @sanity/client.

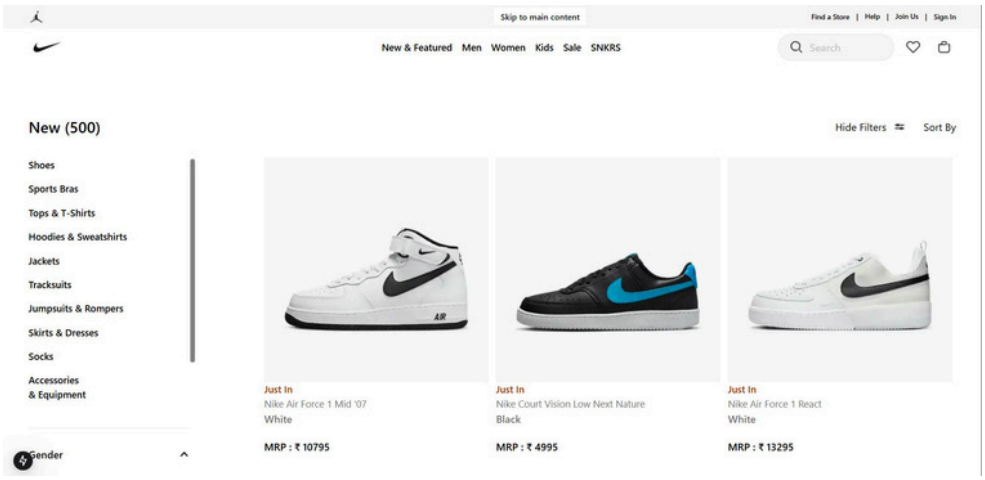
Tools Used:

- Sanity Studio: For schema definition and backend management.
- @sanity/client: For interacting with Sanity backend.
- Next.js: To build API routes for fetching data.

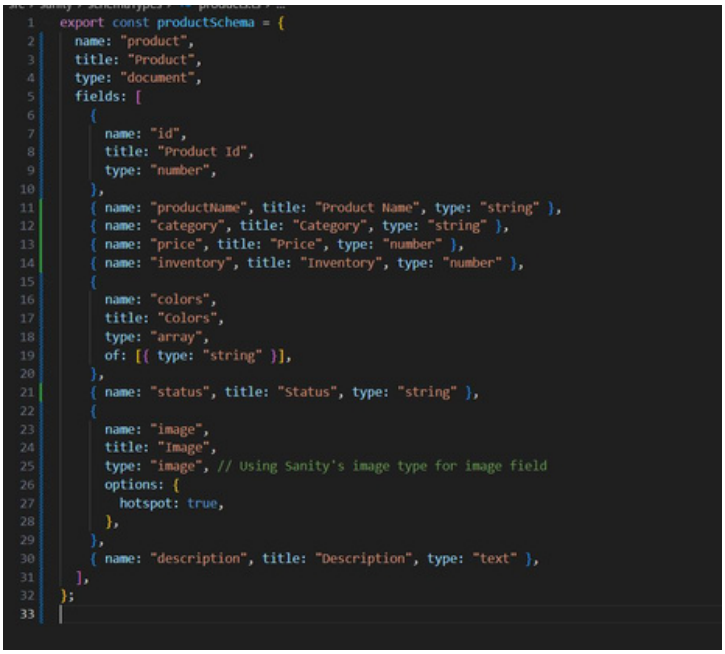
Api Calls

```
const page = () => {

  const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchProducts = async () => {
      setLoading(true);
      setError(null);
      try {
        const res = await fetch("/api/products");
        if (!res.ok) {
          throw new Error("Failed to fetch products");
        }
        const data = await res.json();
        setProducts(data);
      } catch (error) {
        setError(error.message);
      } finally {
        setLoading(false);
      }
    };
    fetchProducts();
  }, []);
```

Frontend



Sanity CMS fields



```
export const productSchema = {
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "id",
      title: "Product Id",
      type: "number",
    },
    { name: "productName", title: "Product Name", type: "string" },
    { name: "category", title: "Category", type: "string" },
    { name: "price", title: "Price", type: "number" },
    { name: "inventory", title: "Inventory", type: "number" },
    {
      name: "colors",
      title: "Colors",
      type: "array",
      of: [{ type: "string" }],
    },
    { name: "status", title: "Status", type: "string" },
    {
      name: "image",
      title: "Image",
      type: "image", // Using Sanity's image type for image field
      options: {
        hotspot: true,
      },
    },
    { name: "description", title: "Description", type: "text" },
  ],
};
```

Code snippets for API integration

```
import { NextResponse } from 'next/server';
import { products } from '../../util/data';

export async function GET() {
    return new Response(JSON.stringify(products), {
        status: 200,
        headers: {
          "Content-Type": "application/json",
          "Cache-Control": "no-store",
          "X-Content-Type-Options": "nosniff",
        },
      });
}
```

Code snippets for Migration Script

```
async function importData() {
    try {
      console.log('migrating data please wait...');

      // Get the products from the API (adjust for correct data struct
      const response = await axios.get('http://localhost:3000/api/dat
      const products = response.data; // Assuming data is inside a "d
      console.log("products ==>> ", products);

      for (const product of products) {
        let imageRef = null;
        if (product.image) {
          imageRef = await uploadImageToSanity(product.image);
        }

        const sanityProduct = {
          _type: 'product',
          id: product.id,
          productName: product.productName,
          category: product.category,
          price: product.price,
          inventory: product.inventory,
          colors: product.colors || [], // Optional, as per your sche
          status: product.status,
          description: product.description,
          image: imageRef ? {
            _type: 'image',
            asset: {
              _type: 'reference',
              _ref: imageRef,
            },
          } : undefined,
        };

        await client.create(sanityProduct);
      }

      console.log('Data migrated successfully!');
    } catch (error) {
      console.error('Error in migrating data ==>> ', error);
    }
  }


importData();
```