Day 3 API Integration and Data Migration

Overview

This is an e-commerce store project where the admin can add products through Sanity.

It integrates with an API to pull product data and uploads it to Sanity for management and display.

The itemShop.tsx component, located inside the myComponents directory, is responsible for handling the d

Features

- Admin Product Management: Admin can add products via Sanity.
- API Integration: The system pulls product data from an external API and uploads it to Sanity.
- Component-Based Design: The data flow is managed within the itemShop.tsx component, which is locate

Learning Outcomes

Throughout the development of this project, you will:

- Learn how to integrate external APIs into your application.
- Understand how to interact with Sanity to manage content and data.
- Gain experience in solving issues and implementing strategies to resolve them during the hackathon.

Installation

1. Clone the repository:

```
git clone https://github.com/your-repository/ecommerce-store.git
```

2. Install dependencies:

```
cd ecommerce-store
npm install
```

3. Set up environment variables for API integration and Sanity.

4. Start the development server:

```
   npm run dev
```

Usage

- Admin can add products through the Sanity interface.
- The API integration will automatically pull product data and upload it to Sanity.
- The itemShop.tsx component is responsible for fetching and uploading product data.

Challenges Faced

- Errors: There were several errors encountered during the development process, each requiring different s
- Data Migration: Ensuring smooth data migration from the API to Sanity was a key challenge and required

Conclusion

This project provided a great opportunity to learn about API integration, content management with Sanity,

and solving complex problems while building a functional e-commerce store.