

Day 3 API Integration and Data Migration

E-Commerce Store Project Overview

This project is an e-commerce store where the admin can easily add products through **Sanity**. It integrates with an external API to pull product data and automatically uploads this data to Sanity for management and display. The core functionality for this process is handled by the `itemShop.tsx` component, located inside the **myComponents** directory of the app folder.

Key Features:

- **Admin Product Management:** The admin has full control to add and manage products directly through the Sanity interface.
 - **API Integration:** The system automatically pulls product data from an external API and uploads it to Sanity for seamless content management.
 - **Component-Based Design:** The `itemShop.tsx` component manages the data flow between the API and Sanity, ensuring efficient data fetching and uploading.
-

Learning Outcomes:

Throughout this project, you'll gain hands-on experience in:

- Integrating external APIs into your application.
 - Interacting with Sanity to effectively manage content and data.
 - Solving complex issues and implementing strategies to resolve them during the development process.
-

Installation:

1. **Clone the repository:**
2. `git clone https://github.com/your-repository/ecommerce-store.git`
3. **Install dependencies:**
4. `cd ecommerce-store`
5. `npm install`
6. **Set up environment variables for API integration and Sanity.**
7. **Start the development server:**

8. npm run dev

Usage:

- Admin can add products directly through the **Sanity** interface.
 - The system will automatically pull product data from the API and upload it to Sanity for management.
 - The itemShop.tsx component is responsible for handling the fetching and uploading of product data.
-

Challenges Faced:

- **Errors:** During the development process, various errors were encountered, each requiring different strategies to resolve.
 - **Data Migration:** Migrating data smoothly from the external API to Sanity was a significant challenge, requiring careful planning and integration.
-

Conclusion:

This project provided valuable learning opportunities, including how to integrate external APIs, manage content with Sanity, and overcome challenges during development. It was a rewarding experience in building a functional and scalable e-commerce store.

This version adds clarity, structure, and emphasizes important points, making it more readable and engaging. Let me know if you'd like any further adjustments!