

Day 3: API Integration & Data Migration

Overview:

Day 3 focuses on API integration and data migration for the FurnitureHub Marketplace.

This involves transferring data from Sanity CMS, setting up schema structures, and ensuring efficient data retrieval using GROQ queries in Next.js.

Key Achievements:

- Implemented a **custom migration script** to transfer data from Sanity CMS to FurnitureHub DB.
- Developed structured **schema** for product data management.
- Integrated **GROQ queries** for fetching and displaying data dynamically.
- Ensured **secure handling** of API keys using environment variables.

Custom Migration Process:

1. **Sanity API Setup:**

- Connected to Sanity CMS using project ID and API token.
- Used secure environment variables for authentication.

2. **Fetching Data from Sanity:**

- Applied **GROQ queries** to retrieve structured content (furniture items, prices, descriptions).
- Optimized data extraction for efficiency.

3. **Mapping & Formatting Data:**

- Converted raw data into a structured format compatible with FurnitureHub's schema.

4. **Saving Data to Database:**

- Inserted data into the database using REST API calls.
- Implemented **error handling** to ensure smooth migration.

FurnitureHub Marketplace - Final Documentation

Client-Side Integration:

- **Server-Side Rendering (SSR):** Used `getServerSideProps` in Next.js for optimized data fetching.
- **Dynamic Routing:** Enabled detailed product views using `/[id]` routes.
- **Responsive UI:** Developed **furniture cards** for mobile and desktop compatibility.

Schema Structure:

- **Title:** Stores furniture item name.
- **Slug:** Unique identifier for product URLs.
- **Description:** Detailed product information.
- **Price:** Numeric value ensuring valid entries.
- **Image:** Stores high-resolution product images.
- **Category:** Associates items with predefined categories.

Security Measures:

- Used `.env` files to secure API tokens and database credentials.
- Prevented exposure of sensitive information in frontend code.

Conclusion:

Day 3 focused on laying a **strong backend foundation** for the marketplace.

With successful API integration and structured data management, the platform is now prepared for further scalability.

Thanks to the structured workflow, future developers can seamlessly extend this implementation.