

# System Architecture Overview

Meridian Furniture will use the following system architecture.



This diagram shows the high-level interactions between the frontend application, the Sanity CMS API, and third-party APIs. The frontend application interacts with the Sanity CMS API to fetch and update content. The Sanity CMS API also interacts with third-party APIs, such as for fetching data or integrating with other services. The arrows indicate the direction of the interactions.

## Key Workflows

The following are the details of the key workflows for the Meridian Furniture Website.

### User Registration

1. User navigates to the registration page on the frontend application.
2. User enters their personal information (name, email, password, etc.) into the registration form.
3. Frontend application sends the user's registration data to the Sanity CMS API.
4. Sanity CMS API validates the user's input and creates a new user account.
5. Sanity CMS API sends a confirmation message back to the frontend application.
6. Frontend application displays a success message to the user.

### Product Browsing

1. User navigates to the product listing page on the frontend application.
2. Frontend application sends a request to the Sanity CMS API to fetch the available products.
3. Sanity CMS API retrieves the product data from its database and sends it back to the frontend application.
4. Frontend application displays the product information (name, description, price, images, etc.) to the user.
5. User can filter, sort, and search the product listings using the frontend application's UI.
6. When the user clicks on a specific product, the frontend application sends a request to the Sanity CMS API to fetch the detailed product information.
7. Sanity CMS API retrieves the detailed product data and sends it back to the frontend application.

8. Frontend application displays the detailed product information to the user.

## Order Placement

1. User adds products to their shopping cart in the frontend application.
2. User proceeds to the checkout page and enters their shipping and payment information.
3. Frontend application sends the order details (products, quantities, shipping, and payment) to the Sanity CMS API.
4. Sanity CMS API validates the order information and creates a new order in its database.
5. Sanity CMS API sends a confirmation message back to the frontend application.
6. Frontend application displays an order confirmation page to the user.
7. Sanity CMS API may also send notifications to third-party APIs (e.g., payment gateway, shipping provider) to process the order.

## API Requirements

Endpoint	Method	Payload	Response
/customers	POST	<pre>{ "name": "John Doe",   "email":     "john@example.com",   "password":     "password123" }</pre>	<pre>{ "id": "abc123",   "name": "John Doe",   "email":     "john@example.com" }</pre>
/products	GET	-	<pre>[{ "id": "prod123",   "name": "Product A",   "description": "This is Product A", "price": 19.99, "image":   "product-a.jpg" }, {   "id": "prod456",   "name": "Product B",   "description": "This is Product B", "price": 24.99, "image":   "product-b.jpg" }]</pre>
/products/{id}	GET	-	<pre>{ "id": "prod123",   "name": "Product A",   "description": "This is Product A", "price": 19.99, "image":   "product-a.jpg",   "category":     "Electronics",   "inventory": 25 }</pre>

Endpoint	Method	Payload	Response
/orders	POST	<pre>{ "userId": "abc123",   "items": [{     "productId": "prod123",     "quantity": 2 }, {     "productId": "prod456",     "quantity": 1 }],   "shippingAddress": "123 Main St, Anytown USA",   "paymentMethod": "visa- 1234" }</pre>	<pre>{ "id": "order123",   "userId": "abc123",   "items": [{     "productId": "prod123",     "quantity": 2 }, {     "productId": "prod456",     "quantity": 1 }],   "shippingAddress": "123 Main St, Anytown USA",   "paymentMethod": "visa- 1234", "total": 54.97,   "status": "pending" }</pre>

The user registration endpoint allows for the creation of a new user account, with the payload containing the user's name, email, and password. The response includes the user's unique identifier, name, and email.

The product listing endpoint retrieves a list of available products, with each product's details such as name, description, price, and image. The individual product endpoint retrieves the detailed information for a specific product, including additional details like category and inventory.

The order placement endpoint allows the user to submit a new order, with the payload containing the user's ID, the items in the order, the shipping address, and the payment method. The response includes the order's unique identifier, the order details, the total amount, and the current status of the order.

These will serve as the requirements for the API which the Meridian frontend will need to serve our customers and ensure a smooth user experience.