FURNITURE E-COMMERCE WEBSITE TECHNICAL FOUNDATION

1. Technical Requirements

Frontend Requirements:

- User-friendly interface for browsing Sofa and Chair furniture products
- Responsive design for mobile and desktop users
- Essential pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation
- Product filtering and search functionality
- User account management

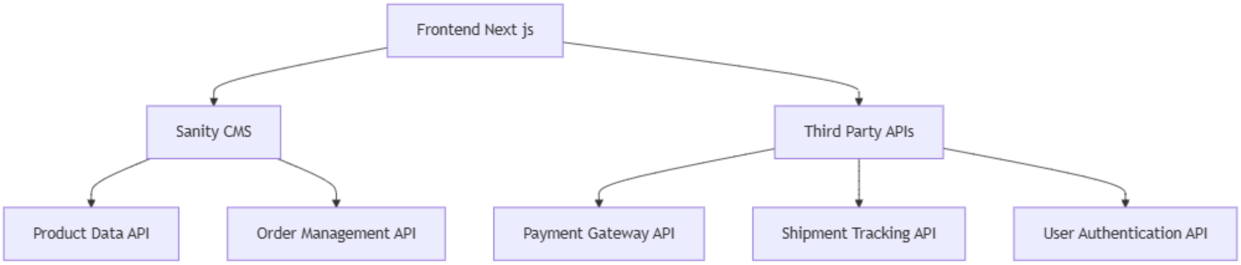
Backend Requirements (Sanity CMS):

- Manage product data (furniture items, categories, prices)
- Store customer details and order records
- Handle inventory management

Third-Party API Integration:

- Payment gateway (e.g., Stripe)
- Shipment tracking (e.g., Shipengine)
- User authentication (e.g., clerk)

2. System Architecture



3. API Requirements for Furniture E-Commerce Website

Product API

Endpoints for managing and retrieving furniture product details:

Endpoint	Method	Description
/api/products	GET	Get a list of all products.
/api/products/:id	GET	Get details of a specific product by ID.

Order API

Endpoints for creating and retrieving customer orders:

Endpoint	Method	Description
/api/orders	POST	Create a new order with product details and shipping address.
/api/orders/:id	GET	Retrieve details of a specific order by ID.

Shipment Tracking API

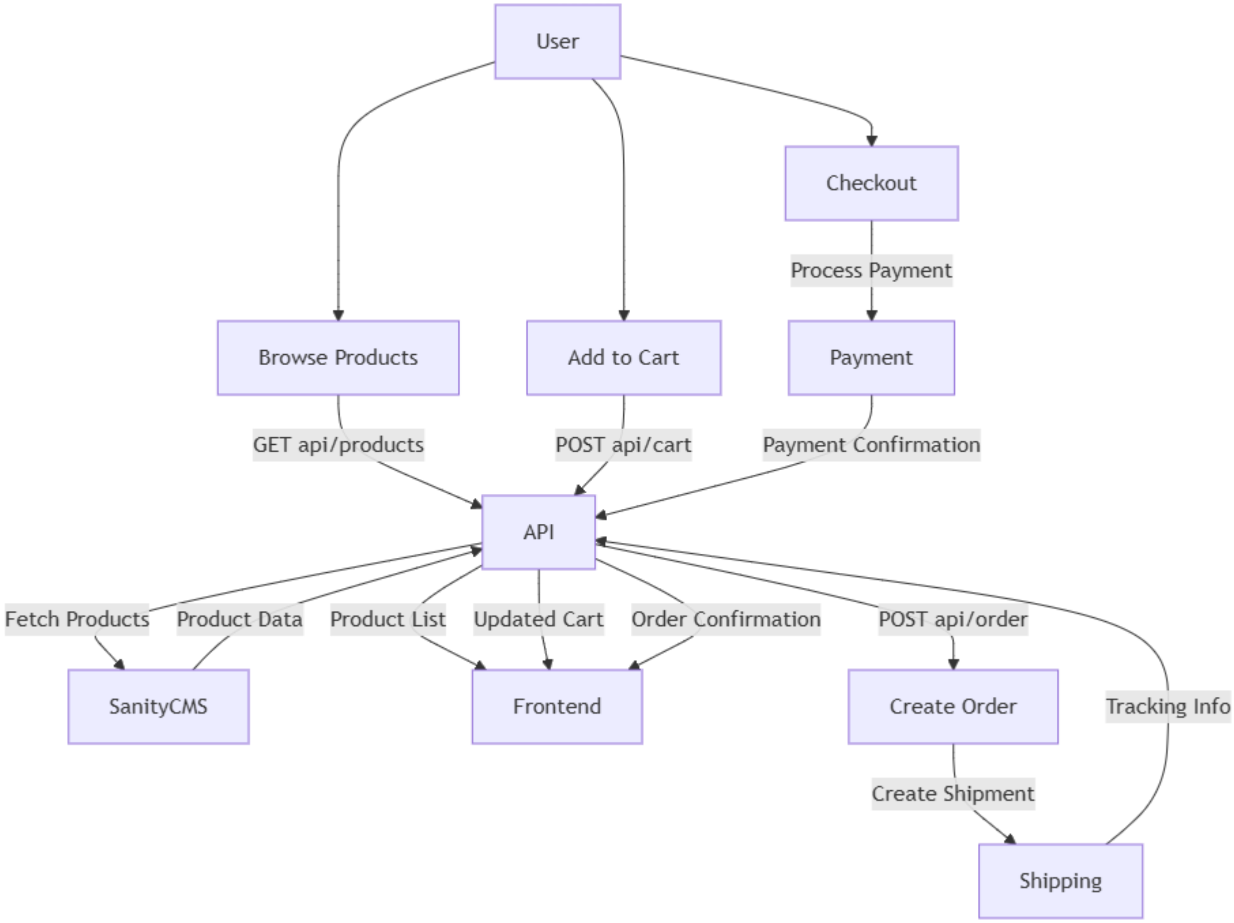
Endpoint for tracking order shipment status:

Endpoint	Method	Description
/api/shipment/:orderId	GET	Get shipment status and estimated delivery time for an order.

Payment API

Endpoint for processing payments:

Endpoint	Method	Description
/api/payment	POST	Process payment for an order with order ID and payment details.

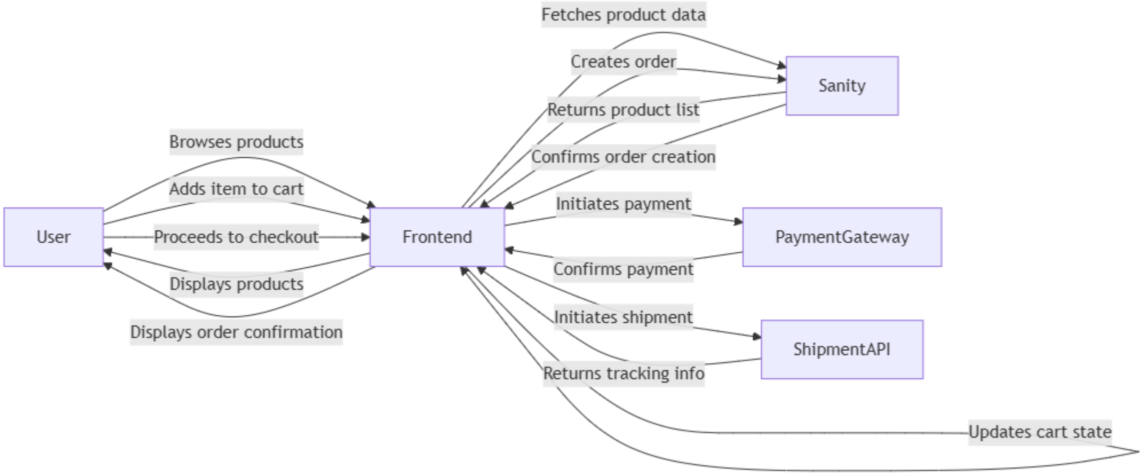


4. Sanity Schema Example

```
// schemas/product.js
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required()
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: {
       source: 'name',
       maxLength: 96
      }
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text'
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
      validation: Rule => Rule.required().min(0)
    },
    {
      name: 'category',
      title: 'Category',
      type: 'reference',
      to: [{type: 'category'}]
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
      options: {
       hotspot: true
      }
    },
    {
      name: 'stock',
      title: 'Stock',
      type: 'number',
      validation: Rule => Rule.required().min(0)
    }
  ]
}

// schemas/order.js
export default {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    {
      name: 'user',
      title: 'User',
      type: 'reference',
      to: [{type: 'user'}]
    },
    {
      name: 'products',
      title: 'Products',
      type: 'array',
      of: [
       {
        type: 'object',
        fields: [
         {
          name: 'product',
          type: 'reference',
          to: [{type: 'product'}]
         },
         {
          name: 'quantity',
          type: 'number'
         }
        ]
       }
      ]
    },
    {
      name: 'totalAmount',
      title: 'Total Amount',
      type: 'number'
    },
    {
      name: 'status',
      title: 'Status',
      type: 'string',
      options: {
       list: ['pending', 'processing', 'shipped', 'delivered']
      }
    },
    {
      name: 'shippingAddress',
      title: 'Shipping Address',
      type: 'text'
    },
    {
      name: 'createdAt',
      title: 'Created At',
      type: 'datetime'
    }
   ]
  }
```

5. Key Workflows


Fetches product data
Creates order
Sanity
Returns product list
Browses products
Confirms order creation
Adds item to cart
Initiates payment
User
Proceeds to checkout
Frontend
PaymentGateway
Displays products
Confirms payment
Displays order confirmation
Initiates shipment
ShipmentAPI
Returns tracking info
Updates cart state