

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

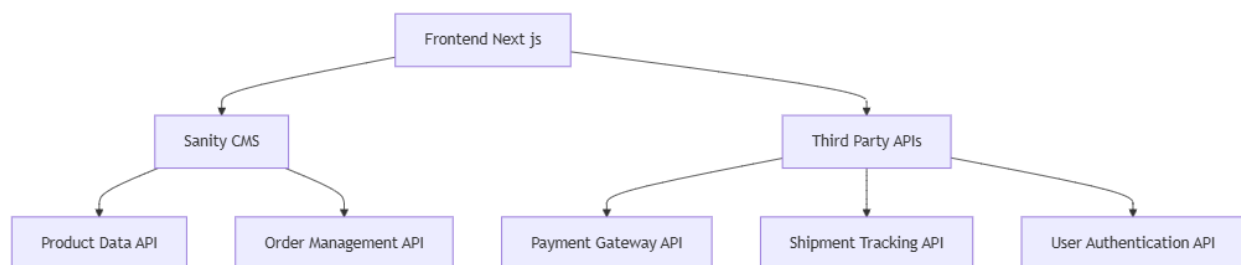
Backend Requirements (Sanity CMS):

- 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., FedEx)
- User authentication (e.g., Auth0)

2. System Architecture



3. API Requirements

Product API

```
interface Product {  
  id: string;  
  name: string;  
  description: string;  
  price: number;  
  category: string;  
  imageUrl: string;  
  stock: number;  
}
```

GET /api/products

Response: Product[]

GET /api/products/:id

Response: Product

Order API

```
interface Order {  
  id: string;  
  userId: string;  
  products: {productId: string, quantity: number}[];  
  totalAmount: number;  
  status: 'pending' | 'processing' | 'shipped' | 'delivered';  
  shippingAddress: string;  
  createdAt: string;  
}
```

POST /api/orders

Request Body: Omit<Order, 'id' | 'status' | 'createdAt'>

Response: Order

GET /api/orders/:id

Response: Order

Shipment Tracking API

GET /api/shipment/:orderId

Response: { orderId: string, status: string, estimatedDelivery: string }

Payment API

POST /api/payment

Request Body: { orderId: string, amount: number, paymentMethod: string }

Response: { success: boolean, transactionId: string }

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

