

TABLE OF CONTENT

- **Introduction**
- **Project Description**
- **Technical Requirements**
- **System Architecture**
- **API Endpoints**
- **Sanity Schema**
- **Conclusion**

INTRODUCTION:

The attached image contains an introduction to a project focused on developing an e-commerce platform. Here's the summarized context:

- **E-commerce Trend:** Online shopping is increasingly common as businesses adopt websites for commercial transactions.
- **Project Objective:** Create a general-purpose e-commerce store for purchasing products from home. The implementation focuses on an online furniture store offering items like wing chairs, armchairs, luxury chairs.
- **Process Overview:** Customers can browse, select products, and add them to a virtual shopping cart. During checkout, they provide details like billing/shipping addresses and payment methods (e.g., EasyPaisa, JazzCash, Cash on Delivery).

PROJECT DESCRIPTION:

- Users can register to view and purchase products.
- User can Contact through E-mail/Contact No on the "Contact us page".
- Here are the rules for user and visitor:
 - ✓ User: Can view and purchase products.
 - ✓ Visitor: Can view, search, and compare products.
- Managing products (add, edit, delete).
- Managing users (add, edit, delete).
- Shipping orders and sending confirmation emails.

TECHNICAL REQUIREMENTS:

Technical Documentation for E-Commerce.

This documentation provides a comprehensive guide to the e-commerce system's architecture, workflows, API endpoints, and Sanity CMS schema examples.

SYSTEM ARCHITECTURE:

- **Frontend :**

The user interface where customers browse the menu, place orders, and track deliveries.

- **Framework:** React/Next.js
- **Styling:** Tailwind CSS
- **Responsive design** for mobile and desktop.
- **Essential Pages & Features:**

- **Homepage:**

Showcase featured products and deals.

- **About**

Share the company's story, mission, and unique craftsmanship.

- **Contact**

Display contact information (address, phone, email) with a contact form and embedded Google Map.

- **Products**

List products with filters (e.g., price, material, color).

- **Shop**

Full catalog showcasing all products with search and cart functionality.

- **Signup**

Allow new users to create accounts.

- **Login**

Enable secure login for existing users.

- **Backend:**

A server that processes the orders, manages the inventory, and integrates the third-party APIs

- **CMS (Sanity):**

For managing dynamic content such as:

- Product Data (Image, Price, Description)
- Customer Data (Profiles, Order History)
- Order Data (Products, Quantities, Status)

➤ Third-Party API Integration:

To enhance the frontend functionality and use experience of the website, the following third party API's will be integrated.

➤ Payment Gateway API:

- **Purpose:** Secure and seamless payment processing for furniture purchases.
- **Examples:** Easy Paisa, Jazz Cash, Razor pay.
- **Features:**
 - Accept credit/debit card and digital wallet payments.
 - Transaction status tracking.

➤ Shipment & Delivery API (Ship Engine)

- **Purpose:** Manage real-time order tracking and delivery logistics.
- **Features:**
 - Generate shipping labels for furniture orders.
 - Track order status (Processing, Shipped, Delivered).
 - Estimate delivery time and shipping costs.
 - Support multiple carriers like Leopards.

API ENDPOINTS:

Endpoint	Method	Description	Parameters	Response
/api/products	GET	Fetch a list of all available furniture items.	category (optional): Filter by category (e.g., chairs, tables).	List of products: id, name, price, category, description, image, availability.
/api/products/{id}	GET	Fetch details of a single product by ID.	id: The unique identifier of the product.	Product details: id, name, price, description, category, image, availability.
/api/cart/add	POST	Add a product to the shopping cart.	productId: ID of the product, quantity: Number of items to add.	Updated cart: items, totalQuantity, totalPrice.
/api/products/{id}	PUT	Update an existing product's details.	id: Product ID, name (optional), price (optional), category (optional), description (optional), image (optional).	Updated product details: id, name, price, category, description, image.
/api/products/{id}	DELETE	Remove a furniture product from the catalog.	id: Product ID.	Confirmation: message (e.g., 'Product deleted successfully').

➤ SANITY SCHEMA:

Product Schema

```
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Product Name',
    },
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    {
      name: 'category',
      type: 'string',
      title: 'Category',
      options: {
        list: [
          { title: 'Arm Chair', value: 'arm_chair' },
          { title: 'Desk Chair', value: 'desk_chair' },
          { title: 'Wing Chair', value: 'wing_chair' },
          { title: 'Park Bench', value: 'park_bench' },
          { title: 'Wooden Chair', value: 'wooden_chair' },
          { title: 'Sofas', value: 'sofas' },
        ],
      },
    },
    {
      name: 'stock',
      type: 'number',
      title: 'Stock',
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
    },
  ],
};
```

2. Order Schema

```
export default {
  name: 'order',
  type: 'document',
  title: 'Order',
```

```

fields: [
  {
    name: 'orderId',
    type: 'string',
    title: 'Order ID',
  },
  {
    name: 'customer',
    type: 'reference',
    to: [{ type: 'customer' }],
    title: 'Customer',
  },
  {
    name: 'products',
    type: 'array',
    title: 'Products Ordered',
    of: [{ type: 'reference', to: [{ type: 'product' }] }],
  },
  {
    name: 'totalAmount',
    type: 'number',
    title: 'Total Amount',
  },
  {
    name: 'orderDate',
    type: 'datetime',
    title: 'Order Date',
  },
  {
    name: 'status',
    type: 'string',
    title: 'Order Status',
    options: {
      list: [
        { title: 'Pending', value: 'pending' },
        { title: 'Shipped', value: 'shipped' },
        { title: 'Delivered', value: 'delivered' },
        { title: 'Cancelled', value: 'cancelled' },
      ],
    },
  },
],
};

```

3. Customer Schema

```

export default {
  name: 'customer',
  type: 'document',
  title: 'Customer',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Customer Name',
    },
    {
      name: 'email',
      type: 'string',
      title: 'Email',
    },
  ],
};

```

```

    name: 'phone',
    type: 'string',
    title: 'Phone Number',
  },
  {
    name: 'address',
    type: 'text',
    title: 'Address',
  },
  {
    name: 'orders',
    type: 'array',
    title: 'Orders',
    of: [{ type: 'reference', to: [{ type: 'order' }] }],
  },
],
};

```

4. Shipment Schema

```

export default {
  name: 'shipment',
  type: 'document',
  title: 'Shipment',
  fields: [
    {
      name: 'shipmentId',
      type: 'string',
      title: 'Shipment ID',
    },
    {
      name: 'order',
      type: 'reference',
      to: [{ type: 'order' }],
      title: 'Order',
    },
    {
      name: 'carrier',
      type: 'string',
      title: 'Carrier Name',
    },
    {
      name: 'trackingNumber',
      type: 'string',
      title: 'Tracking Number',
    },
    {
      name: 'status',
      type: 'string',
      title: 'Shipment Status',
      options: {
        list: [
          { title: 'In Transit', value: 'in_transit' },
          { title: 'Delivered', value: 'delivered' },
          { title: 'Pending', value: 'pending' },
          { title: 'Cancelled', value: 'cancelled' },
        ],
      },
    },
    {
      name: 'estimatedDelivery',
      type: 'datetime',
    },
  ],
};

```

```
        title: 'Estimated Delivery',  
      },  
    ],  
  };
```

CONCLUSION:

The project ensures a seamless shopping experience by integrating modern technologies such as Sanity CMS and Next.js. By focusing on an intuitive user journey, efficient order processing, and robust API integration, it delivers a streamlined e-commerce solution. The implementation roadmap outlines clear phases for schema design, frontend development, and API integration, ensuring efficient development and deployment.