

## DOCUMENTAION OF MARKETPLACE E-COMMERCE WEBSITE

### TECHINCAL REQUIREMENTS

#### **Frontend Requirements: (NEXT.JS)**

- User-friendly interface for browsing products.
- Responsive design for mobile and desktop users.
- Essential pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation

#### **Sanity CMS as Backend:**

```
// schemas/order.js
```

```
export default {  
  name: 'order',  
  title: 'Order',  
  type: 'document',  
  fields: [  
    {  
      name: 'customerName',  
      title: 'Customer Name',  
      type: 'string',  
    },  
    {  
      name: 'product',  
      title: 'Product',  
      type: 'reference',  
      to: [{ type: 'product' }],  
    },  
    {  
      name: 'quantity',  
      title: 'Quantity',  
      type: 'number',  
    },  
  ],  
}
```

```
{
  name: 'status',
  title: 'Order Status',
  type: 'string',
  options: {
    list: [
      { title: 'Pending', value: 'pending' },
      { title: 'Shipped', value: 'shipped' },
      { title: 'Delivered', value: 'delivered' },
    ],
  },
},
{
  name: 'paymentDetails',
  title: 'Payment Details',
  type: 'object',
  fields: [
    {
      name: 'paymentMethod',
      title: 'Payment Method',
      type: 'string',
      options: {
        list: [
          { title: 'Credit Card', value: 'credit_card' },
          { title: 'PayPal', value: 'paypal' },
          { title: 'Cash on Delivery', value: 'cod' },
        ],
      },
    },
  ],
},
```

```
{
  name: 'paymentStatus',
  title: 'Payment Status',
  type: 'string',
  options: {
    list: [
      { title: 'Pending', value: 'pending' },
      { title: 'Paid', value: 'paid' },
      { title: 'Failed', value: 'failed' },
    ],
  },
},
{
  name: 'transactionId',
  title: 'Transaction ID',
  type: 'string',
},
{
  name: 'amount',
  title: 'Amount',
  type: 'number',
},
],
},
],
};
```

### ***Third-Party APIs:***

I'll use two third party APIs

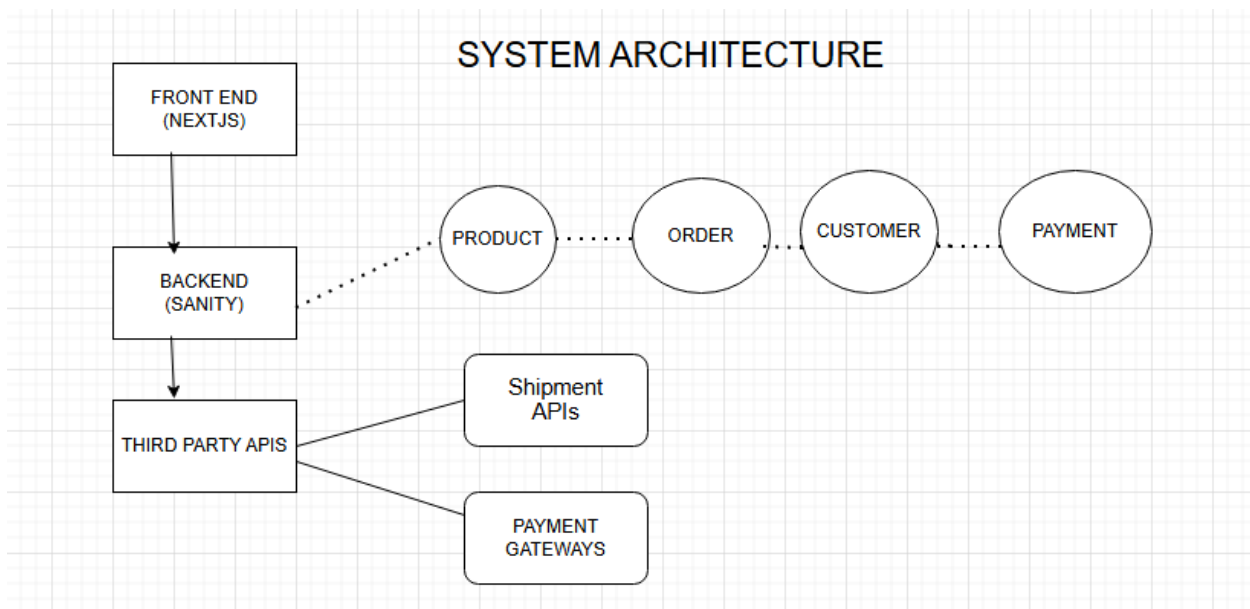
1. Shipment APIs:

Shipment APIs allow me to integrate courier and logistics services into my application. With these APIs, I can automate tasks such as creating shipments, tracking parcels, calculating shipping rates, and managing returns (e.g shippo, BHL etc)

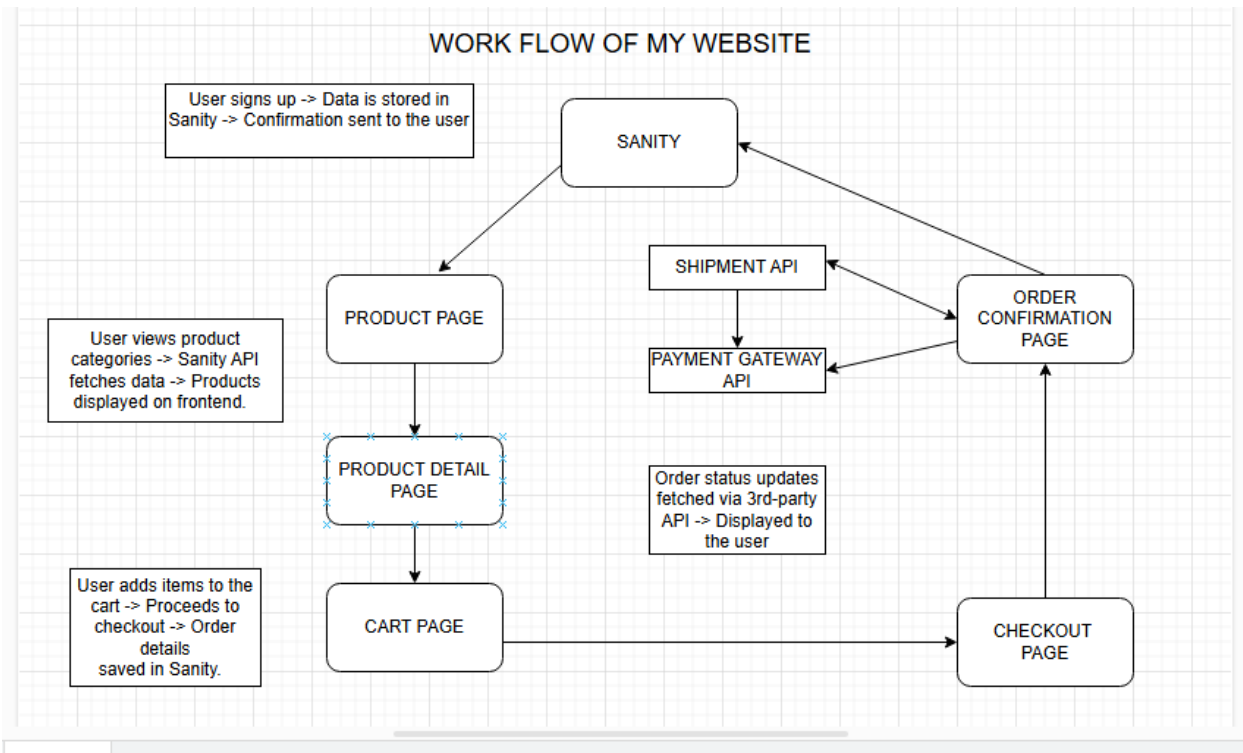
2. Payment gateways:

Payment gateway APIs facilitate secure payment processing on your website or app. They allow you to accept payments via credit/debit cards, wallets, and other online payment methods.(e.g stripe, paypal etc)

### ***System architecture design:***



### Key workflow:



### Plan API requirements:

1. Endpoint: /products

Method: get

Description: fetch all data

Response example:

```
[
  {
    "id": "p001",
    "name": "Rose Elegance",
    "description": "A floral scent with hints of rose and jasmine.",
    "price": 50.99,
    "tags": ["floral", "luxury"],
    "reviews": 4.5,
    "image": "https://example.com/images/rose-elegance.jpg"
  }
]
```

2. Endpoint: /order

Method: post

Description: create a new order

Payload:

```
{
  "customerName": "Hamza Javed",
  "productId": "p001",
  "quantity": 2
}
```

} Response example:

```
{
  "message": "Order placed successfully.",
  "orderId": "o002",
  "totalPrice": 101.98
}
```

Endpoint: /payment

Method: post

Description: Process a payment for an order.

Payload:

```
{
  "orderId": "o001",
  "amount": 101.98,
  "paymentMethod": "credit_card"
}
```

Response example:

```
{
  "message": "Payment processed successfully.",
  "paymentId": "pay002",
  "transactionId": "txn_456DEF",
  "paymentStatus": "paid"
}
```

Endpoint: /shipment

Method: get

Description: track the shipment status for an order

Response example :

```
[
  {
    "shipmentId": "sh001",
    "orderId": "o001",
    "carrierName": "FedEx",
    "trackingNumber": "FDX12345678",
    "trackingStatus": "In Transit",
    "lastUpdated": "2025-01-15T10:30:00Z"
  }
]
```

