SHOP.CO Sanity CMS Schema Documentation

This document outlines the detailed and structured schema definitions for the SHOP.CO e-commerce platform, developed using **Sanity CMS**. It describes the key content types for the platform: **Users**, **Products**, **Orders**, **Payments**, and **Shipping**. These schemas will facilitate the seamless management of product listings, customer information, orders, payments, and shipping details in an efficient, real-time, and secure manner.

1. Product Schema

The **Product** schema captures all necessary details regarding the product, such as product name, description, price, stock, images, and categories.

Schema Definition

```
// schemas/product.js
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required().min(1).max(100)
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: {
        source: 'name',
        maxLength: 96,
      },
      validation: Rule => Rule.required()
    },
    {
```

```
  name: 'price',
  title: 'Price',
  type: 'number',
  validation: Rule => Rule.required().min(0)
},
{
  name: 'stock',
  title: 'Stock',
  type: 'number',
  validation: Rule => Rule.required().min(0)
},
{
  name: 'description',
  title: 'Description',
  type: 'text',
  validation: Rule => Rule.required().min(20)
},
{
  name: 'images',
  title: 'Images',
  type: 'array',
  of: [{ type: 'image' }],
  options: {
    hotspot: true
  }
},
{
  name: 'categories',
  title: 'Categories',
  type: 'array',
  of: [
    {
      type: 'reference',
      to: [{ type: 'category' }]
    }
  ],
  validation: Rule => Rule.required().min(1)
},
{
  name: 'createdAt',
```

```
      title: 'Created At',
      type: 'datetime',
      initialValue: () => new Date().toISOString(),
    }
  ]
}
```

Key Fields:

- **name**: Product name.
- **slug**: URL-friendly identifier generated from the product name.
- **price**: Price of the product.
- **stock**: Number of items available.
- **description**: Detailed product description.
- **images**: Array of images related to the product.
- **categories**: Categories assigned to the product (such as Men's Fashion, Women's Fashion).
- **createdAt**: Timestamp of product creation.

---

2. Category Schema

The **Category** schema organizes products into categories for easier navigation and filtering on the platform. Categories are used to group similar items together, such as Men's Fashion, Women's Fashion, and Kids' Clothing.

Schema Definition

```
// schemas/category.js
export default {
  name: 'category',
  title: 'Category',
  type: 'document',
```

```
  fields: [
    {
      name: 'name',
      title: 'Name',
      type: 'string',
      validation: Rule => Rule.required().min(1).max(50)
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: {
        source: 'name',
        maxLength: 96,
      },
      validation: Rule => Rule.required()
    }
  ]
}
```

Key Fields:

- **name**: Category name (e.g., Men's Fashion, Women's Fashion).
- **slug**: URL-friendly category identifier.

---

3. User Schema

The **User** schema captures customer data, including personal details and order history. It also stores user roles to differentiate between customers and administrators.

Schema Definition

```
// schemas/user.js
export default {
  name: 'user',
  title: 'User',
  type: 'document',
  fields: [
```

```
{
  name: 'firstName',
  title: 'First Name',
  type: 'string',
  validation: Rule => Rule.required().min(1).max(50)
},
{
  name: 'lastName',
  title: 'Last Name',
  type: 'string',
  validation: Rule => Rule.required().min(1).max(50)
},
{
  name: 'email',
  title: 'Email',
  type: 'string',
  validation: Rule => Rule.required().email()
},
{
  name: 'role',
  title: 'Role',
  type: 'string',
  options: {
    list: [
      { title: 'Customer', value: 'customer' },
      { title: 'Admin', value: 'admin' }
    ],
  },
  validation: Rule => Rule.required()
},
{
  name: 'orders',
  title: 'Orders',
  type: 'array',
  of: [
    {
      type: 'reference',
      to: [{ type: 'order' }]
    }
  ]
```

```
    }
  ]
}
```

Key Fields:

- **firstName**: User's first name.
- **lastName**: User's last name.
- **email**: User's email address (unique).
- **role**: Role of the user (Customer or Admin).
- **orders**: List of orders made by the user, referenced from the **Order** schema.

---

4. Order Schema

The **Order** schema tracks each order made on SHOP.CO, including the user who placed the order, the products ordered, payment status, shipping details, and order status.

Schema Definition

```
// schemas/order.js
export default {
  name: 'order',
  title: 'Order',
  type: 'document',
  fields: [
    {
      name: 'user',
      title: 'User',
      type: 'reference',
      to: [{ type: 'user' }],
      validation: Rule => Rule.required()
    },
    {
      name: 'products',
      title: 'Products',
      type: 'array',
      of: [
        {
```

```
      type: 'object',
      fields: [
        {
          name: 'product',
          title: 'Product',
          type: 'reference',
          to: [{ type: 'product' }]
        },
        {
          name: 'quantity',
          title: 'Quantity',
          type: 'number',
          validation: Rule => Rule.required().min(1)
        }
      ]
    }
  ]
},
{
  name: 'status',
  title: 'Status',
  type: 'string',
  options: {
    list: [
      { title: 'Pending', value: 'pending' },
      { title: 'Shipped', value: 'shipped' },
      { title: 'Delivered', value: 'delivered' },
      { title: 'Cancelled', value: 'cancelled' }
    ]
  },
  validation: Rule => Rule.required()
},
{
  name: 'totalAmount',
  title: 'Total Amount',
  type: 'number',
  validation: Rule => Rule.required().min(0)
},
{
  name: 'paymentStatus',
```

```
      title: 'Payment Status',
      type: 'string',
      options: {
        list: [
          { title: 'Paid', value: 'paid' },
          { title: 'Pending', value: 'pending' },
          { title: 'Failed', value: 'failed' }
        ]
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'shippingAddress',
      title: 'Shipping Address',
      type: 'object',
      fields: [
        { name: 'address', title: 'Address', type: 'string' },
        { name: 'city', title: 'City', type: 'string' },
        { name: 'postalCode', title: 'Postal Code', type:
'string' },
        { name: 'country', title: 'Country', type: 'string' }
      ]
    },
    {
      name: 'createdAt',
      title: 'Created At',
      type: 'datetime',
      initialValue: () => new Date().toISOString(),
    }
  ]
}
```

Key Fields:

- **user**: Reference to the user who placed the order.
- **products**: List of products with quantity ordered.
- **status**: Current status of the order (e.g., Pending, Shipped).
- **totalAmount**: Total order amount.
- **paymentStatus**: Payment status (Paid, Pending, Failed).

- **shippingAddress**: Shipping address for the order.
- **createdAt**: Timestamp of when the order was placed.

---

5. Payment Schema

The **Payment** schema manages the payment status of each order. It stores information such as the payment method, amount, and payment status.

Schema Definition

```
// schemas/payment.js
export default {
  name: 'payment',
  title: 'Payment',
  type: 'document',
  fields: [
    {
      name: 'order',
      title: 'Order',
      type: 'reference',
      to: [{ type: 'order' }],
      validation: Rule => Rule.required()
    },
    {
      name: 'paymentMethod',
      title: 'Payment Method',
      type: 'string',
      options: {
        list: [
          { title: 'Credit Card', value: 'creditCard' },
          { title: 'PayPal', value: 'paypal' },
          { title: 'Stripe', value: 'stripe' }
        ],
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'amount',
```

```
      title: 'Amount',
      type: 'number',
      validation: Rule => Rule.required().min(0)
    },
    {
      name: 'paymentStatus',
      title: 'Payment Status',
      type: 'string',
      options: {
        list: [
          { title: 'Completed', value: 'completed' },
          { title: 'Pending', value: 'pending' },
          { title: 'Failed', value: 'failed' }
        ]
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'paymentDate',
      title: 'Payment Date',
      type: 'datetime',
      initialValue: () => new Date().toISOString(),
    }
  ]
}
```

Key Fields:

- **order**: Reference to the order for which the payment is made.
- **paymentMethod**: Method of payment (e.g., Stripe, PayPal).
- **amount**: Payment amount.
- **paymentStatus**: Payment status (e.g., Completed, Pending).
- **paymentDate**: Date and time when the payment was processed.

6. Shipping Schema

The **Shipping** schema captures shipping-related details for each order. It includes information about the shipping method, tracking number, status of the shipment, and expected delivery date.

Schema Definition

```
// schemas/shipping.js
export default {
  name: 'shipping',
  title: 'Shipping',
  type: 'document',
  fields: [
    {
      name: 'order',
      title: 'Order',
      type: 'reference',
      to: [{ type: 'order' }],
      validation: Rule => Rule.required()
    },
    {
      name: 'shippingMethod',
      title: 'Shipping Method',
      type: 'string',
      options: {
        list: [
          { title: 'Standard Shipping', value: 'standard' },
          { title: 'Express Shipping', value: 'express' },
          { title: 'Overnight Shipping', value: 'overnight' }
        ]
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'trackingNumber',
      title: 'Tracking Number',
      type: 'string',
      validation: Rule => Rule.required().min(10)
    },
    {
```

```
      name: 'shippingStatus',
      title: 'Shipping Status',
      type: 'string',
      options: {
        list: [
          { title: 'Pending', value: 'pending' },
          { title: 'Shipped', value: 'shipped' },
          { title: 'In Transit', value: 'inTransit' },
          { title: 'Delivered', value: 'delivered' },
          { title: 'Returned', value: 'returned' }
        ]
      },
      validation: Rule => Rule.required()
    },
    {
      name: 'estimatedDeliveryDate',
      title: 'Estimated Delivery Date',
      type: 'datetime',
      validation: Rule => Rule.required()
    },
    {
      name: 'shippingAddress',
      title: 'Shipping Address',
      type: 'object',
      fields: [
        { name: 'address', title: 'Address', type: 'string' },
        { name: 'city', title: 'City', type: 'string' },
        { name: 'postalCode', title: 'Postal Code', type:
'string' },
        { name: 'country', title: 'Country', type: 'string' }
      ]
    },
    {
      name: 'shippedAt',
      title: 'Shipped At',
      type: 'datetime',
      initialValue: () => new Date().toISOString(),
    }
  ]
}
```

Key Fields:

- **order**: Reference to the associated order for which the shipping details are provided.
- **shippingMethod**: Method of shipment (e.g., Standard Shipping, Express Shipping).
- **trackingNumber**: Unique tracking number provided by the shipping carrier.
- **shippingStatus**: Current status of the shipment (e.g., Pending, Shipped, In Transit, Delivered, Returned).
- **estimatedDeliveryDate**: Estimated date for delivery.
- **shippingAddress**: Address to which the order is shipped.
- **shippedAt**: Timestamp of when the order was shipped.