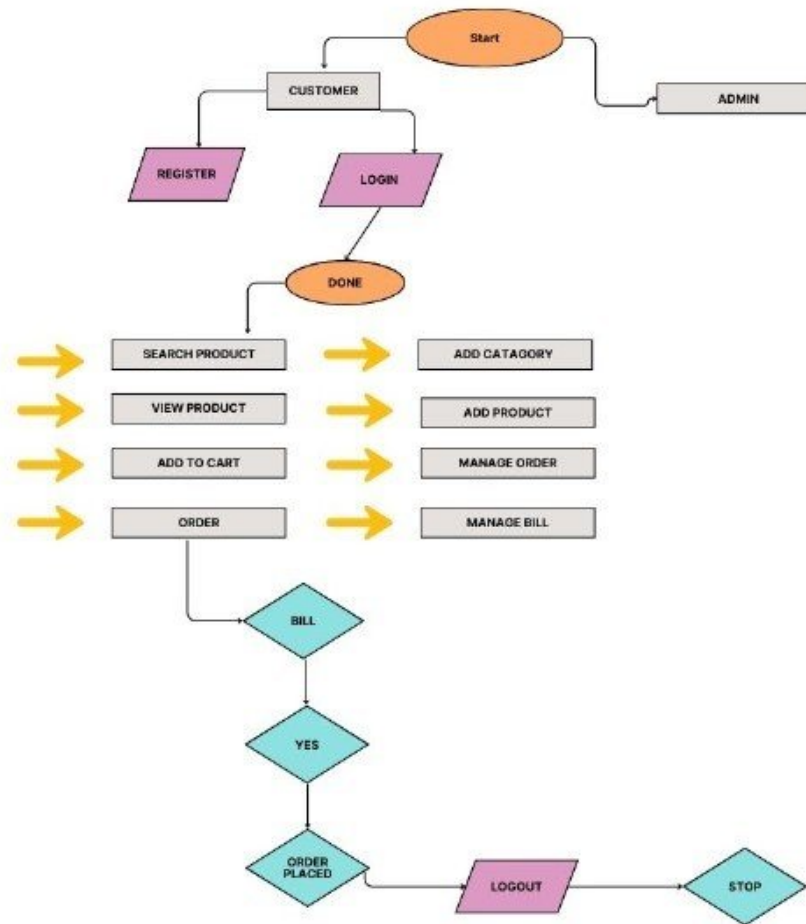


# PLANNING THE TECHNICAL FOUNDATION

## DAY: 2

### EVENTS FOR APPARAL AND GARMENT WEBSITE

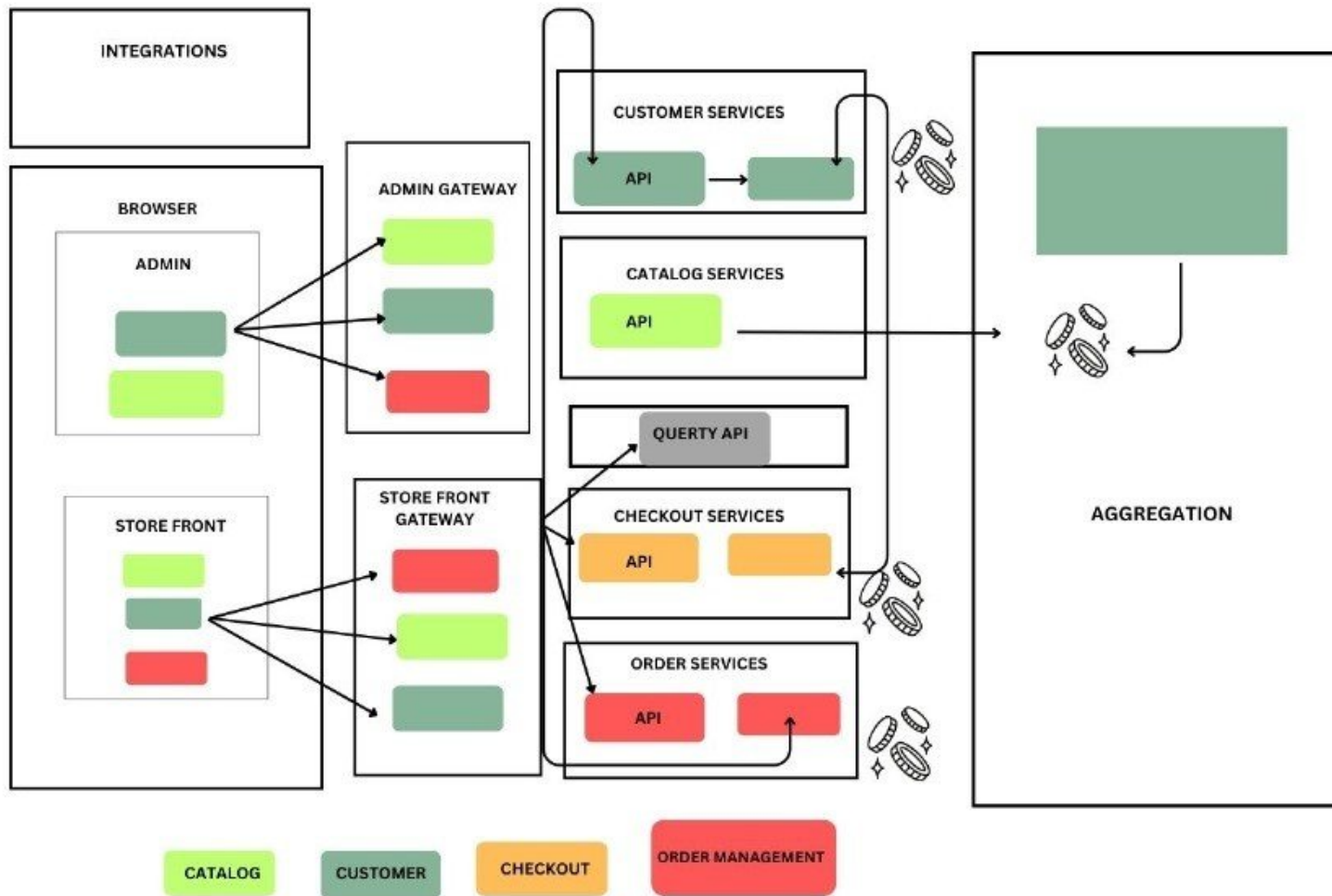


## SYSTEM ARCHITECTURE OVERVIEW

DIAGRAM:



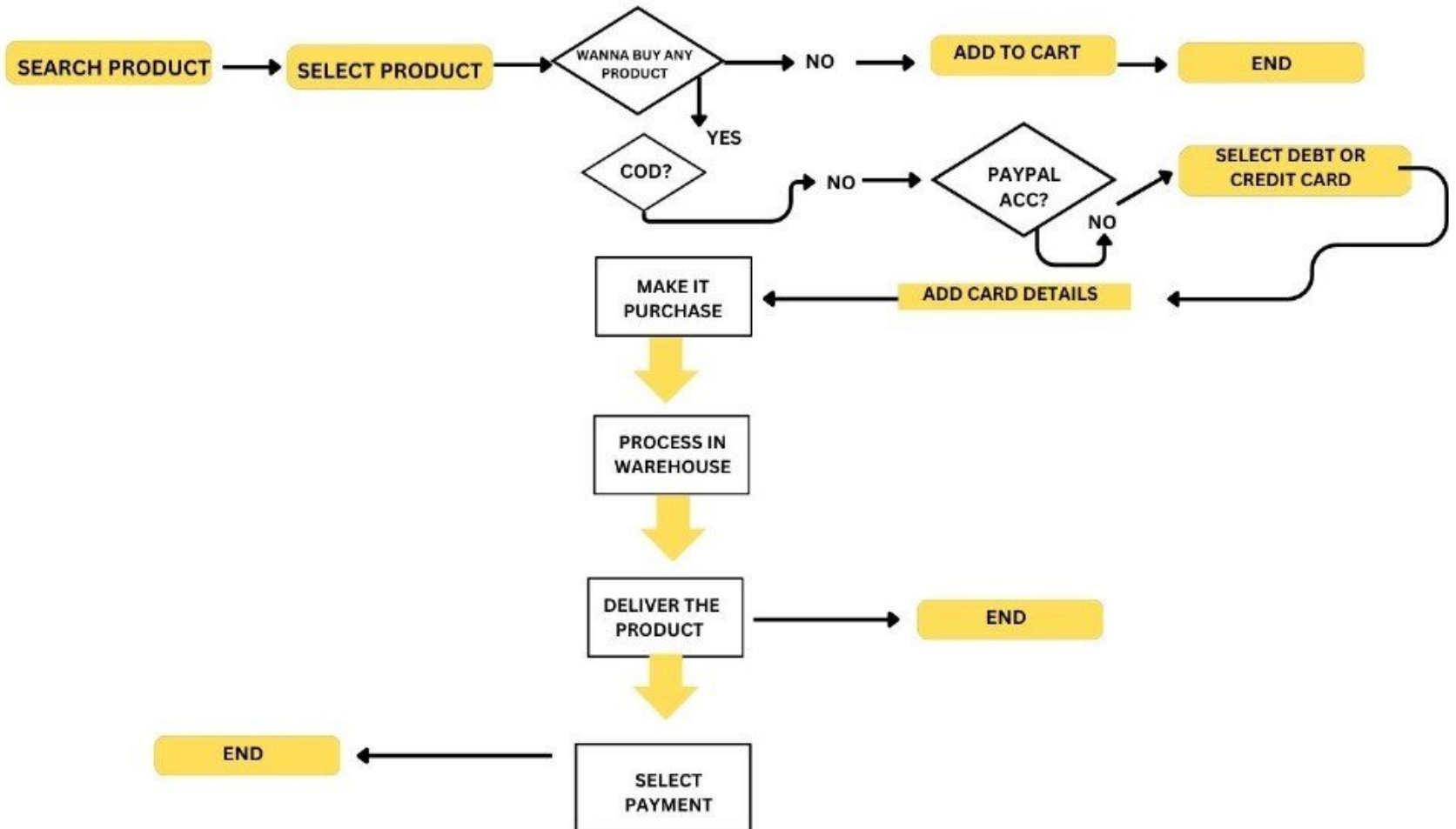
## SYSTEM ARCHITECTUR FLOW CHART



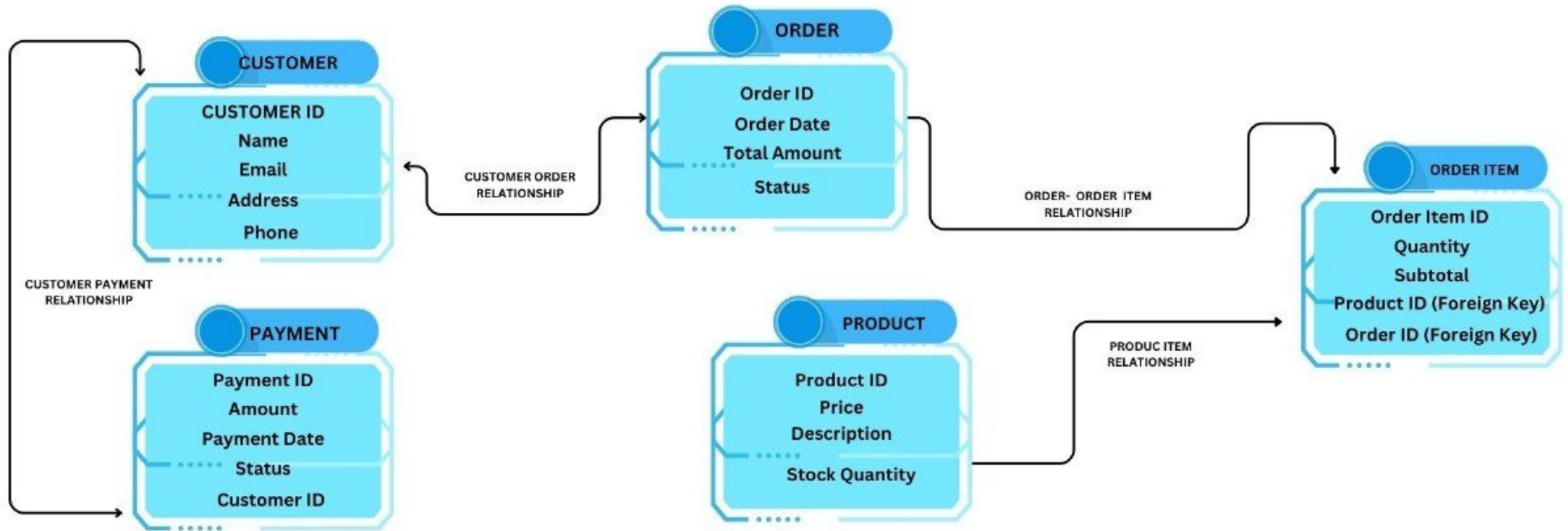
E COMMERCE SITE

PLACING ORDER

SHIPPING ORDER

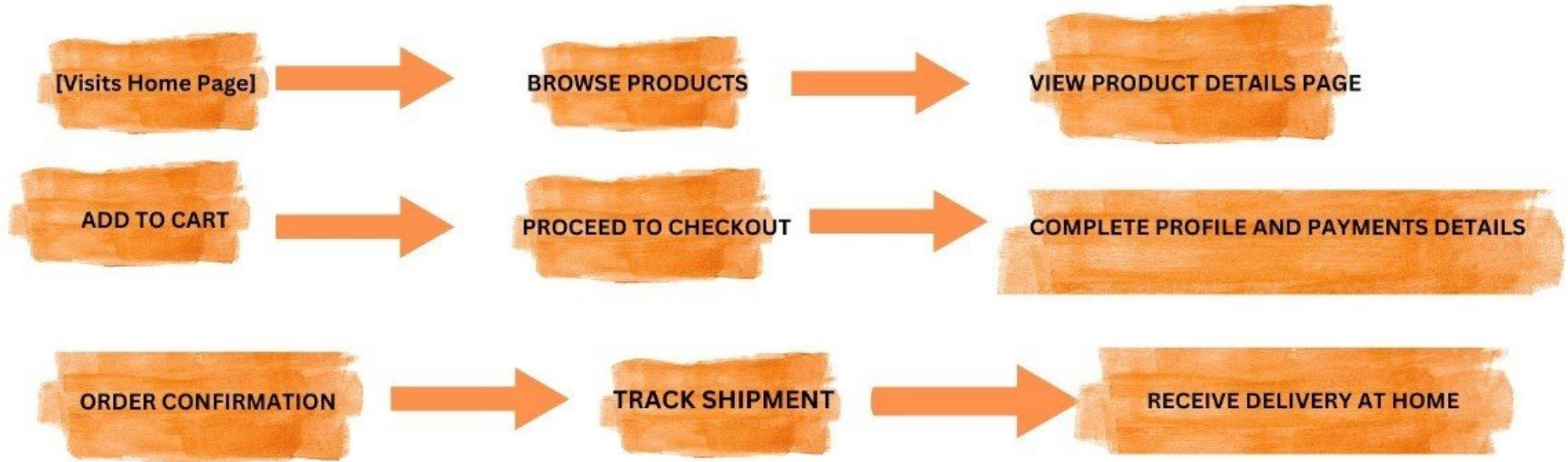


## ER DIADRAM SAMPLE FOR ECOMMERCE PROJECT



API ENDpoints.XLSX				
ENDPOINTS	METHOD	DESCRIPTION	PARAMETERS	RESPONSE EXAMPLE
/api/products	GET	Fetch all products	None	{ id: 1, name: "T-shirt", category: "Tops", price: 29.99 }
/api/products/:id	GET	Fetch a single product	id (Path)	{ id: 1, name: "T-shirt", category: "Tops", price: 29.99 }
/api/products	POST	Add a new product	name, category, price, description, image (Body)	{ success: true, id: 5 }
/api/products/:id	PUT	Update a product	id (Path), name, category, price, description, image (Body)	{ success: true }
/api/products/:id	DELETE	Delete a product	id (Path)	{ success: true }
/api/categories	GET	Fetch all product categories	None	{ categories: ["Tops", "Bottoms", "Dresses", "Accessories"] }

## WORKFLOW DIAGRAM VISUAL REPRESENTATION





## SANITY SCHEMA.js

```

export default {
  // Define the document type and its name
  name: 'product',
  type: 'document',
  title: 'Apparel Product', // The title displayed in the CMS for this document
  fields: [
    // Field for the product's name
    {
      name: 'name',
      type: 'string', // Back: text field
      title: 'Product Name', // Label for the field in the CMS
      validation: (Rule) =>
        Rule.required() // Makes this field mandatory
        .max(100) // Restricts the maximum length to 100 characters
        .error('Product name is required and cannot exceed 100 characters.'),
    },
    // Slug field for generating a URL-friendly identifier
    {
      name: 'slug',
      type: 'slug',
      title: 'Slug',
      description: 'URL-friendly identifier for the apparel product', // Helper text in the CMS
      options: {
        source: 'name', // Automatically generates the slug based on the product name
        maxLength: 200, // Limits the slug length
      },
      validation: (Rule) =>
        Rule.required().error('Slug is required for product identification.'),
    },
    // Field for a detailed description of the product
    {
      name: 'description',
      type: 'text', // Multi-line text field
      title: 'Description',
      description: 'Detailed description of the apparel product',
      validation: (Rule) =>
        Rule.required() // Makes this field mandatory
        .min(20) // Requires at least 20 characters
        .max(500) // Limits to a maximum of 500 characters
        .error('Description must be between 20 and 500 characters.'),
    },
    // Field for product category (e.g., 'Tops', 'Bottoms', 'Dresses')
    {
      name: 'category',
      type: 'string',
      title: 'Category',
      options: {
        list: [
          { title: 'Tops', value: 'Tops' },
          { title: 'Bottoms', value: 'Bottoms' },
          { title: 'Dresses', value: 'Dresses' },
          { title: 'Accessories', value: 'Accessories' },
        ],
        // Add more categories as needed
      },
      validation: (Rule) => Rule.required().error('Category is required.'),
    },
    // Field for product size (e.g., 'S', 'M', 'L', 'XL')
    {
      name: 'size',
      type: 'array',
      of: [{ type: 'string' }],
      title: 'Size(s)',
      options: {
        layout: 'tags', // Display sizes as tags
      },
    },
    // Field for product color
    {
      name: 'color',
      type: 'string',
      title: 'Color',
    },
    // Field for product price
    {
      name: 'price',
      type: 'number',
      title: 'Price',
    },
    validation: (Rule) => Rule.required().min(0).error('Price must be a positive number.'),
  ],
  // Field for product images
  {
    name: 'images',
    type: 'array',
    of: [{ type: 'image' }],
    title: 'Image(s)',
  },
  // ... other fields ...
}

```

```

{
  // Field for the product's price
  name: 'price',
  type: 'number', // Numeric field
  title: 'Product Price',
  validation: (Rule) =>
    Rule.required() // Makes this field mandatory
    .min(0) // Ensures the price is non-negative
    .error('Product price must be a positive value.'),
},
{
  // Field for the discount percentage
  name: 'discountPercentage',
  type: 'number',
  title: 'Discount Percentage',
  description: 'Percentage discount on the apparel product',
  validation: (Rule) =>
    Rule.min(0) // Ensures the discount is not negative
    .max(100) // Ensures the discount does not exceed 100%
    .error('Discount percentage must be between 0 and 100.'),
},
{
  // Field for the price before the discount
  name: 'priceWithoutDiscount',
  type: 'number',
  title: 'Price Without Discount',
  description: 'Original price of the apparel product before discount',
  readOnly: true, // Makes this field non-editable in the CMS
  // Optional: Auto-calculate this field based on price and discount
  initialValue: (doc) => doc.price / (1 - doc.discountPercentage / 100),
},
{
  // Field for the product's rating
  name: 'rating',
  type: 'number',
  title: 'Rating',
  description: 'Average rating of the apparel product',
  validation: (Rule) =>
    Rule.min(0) // Ensures the rating is not negative
    .max(5) // Limits the rating to a maximum of 5
    .precision(1) // Allows one decimal place
    .error('Rating must be between 0 and 5.'),
}

```