POD (Print On Demand) System Documentation
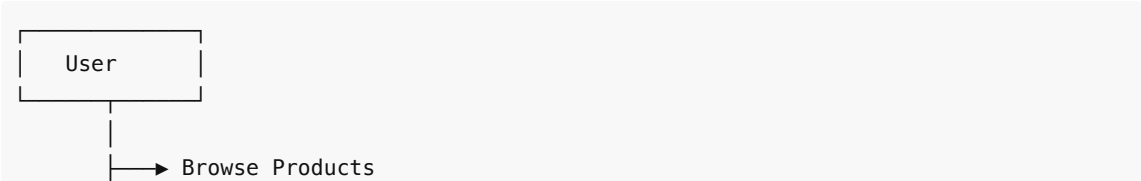
Table of Contents

Overview

The POD (Print On Demand) System is a comprehensive e-commerce solution for custom-printed products. It enables users to browse products, customize designs, add items to cart, place orders, track deliveries, and manage their purchases. The system supports various product categories, sizes, colors, custom designs, express delivery, and cashback rewards.
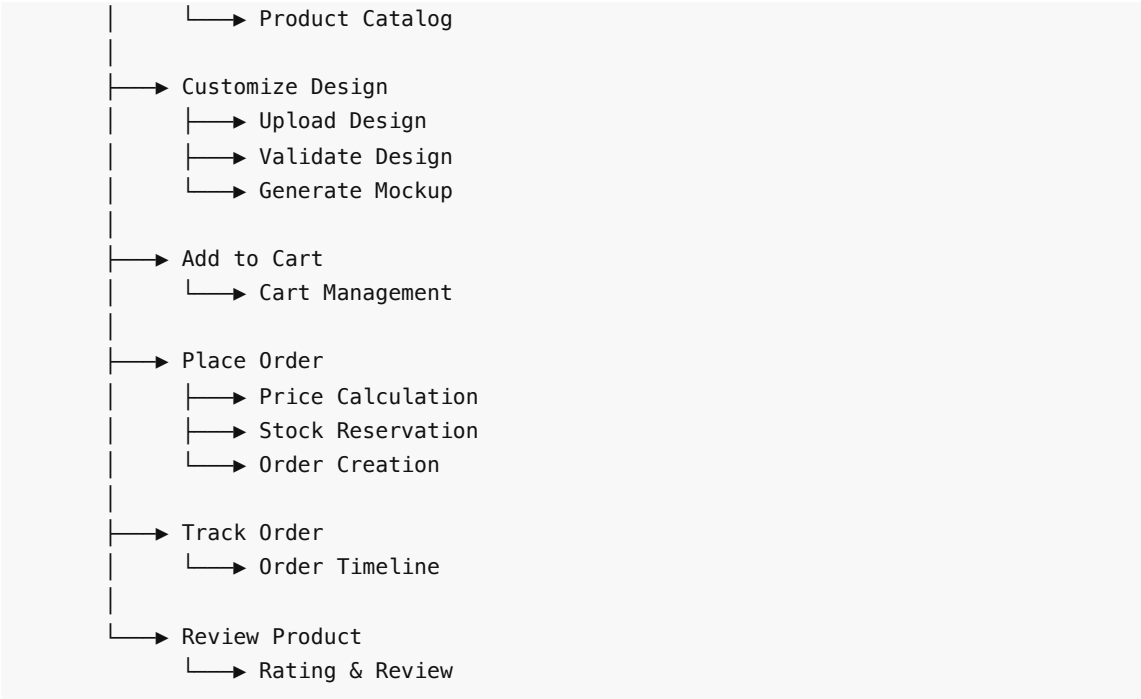
Key Features

- ✅ **Product Catalog**: Browse categories, subcategories, and products with filters
- ✅ **Shopping Cart**: Add, update, and manage cart items
- ✅ **Order Management**: Place orders, track status, cancel, and return
- ✅ **Design Upload**: Upload and validate custom designs
- ✅ **Mockup Generation**: Generate product mockups with designs
- ✅ **Delivery Options**: Standard and express delivery with slot booking
- ✅ **Reviews & Ratings**: Product reviews and ratings system
- ✅ **Coupons & Discounts**: Apply discount coupons
- ✅ **Cashback System**: Earn cashback on purchases
- ✅ **Stock Management**: Real-time stock checking and reservation
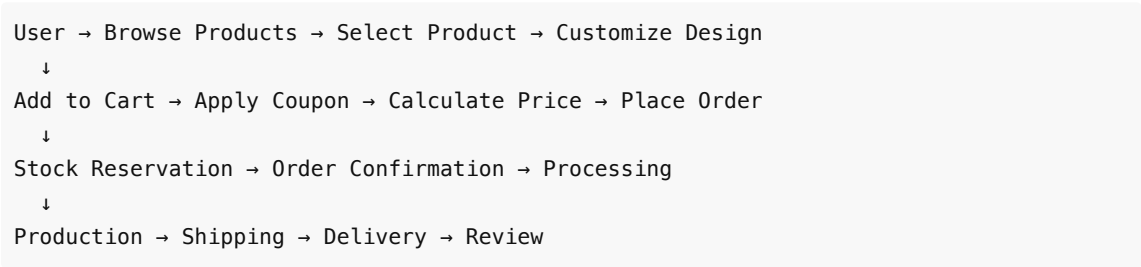- ✅ **Price Calculation**: Automatic pricing with discounts and delivery charges
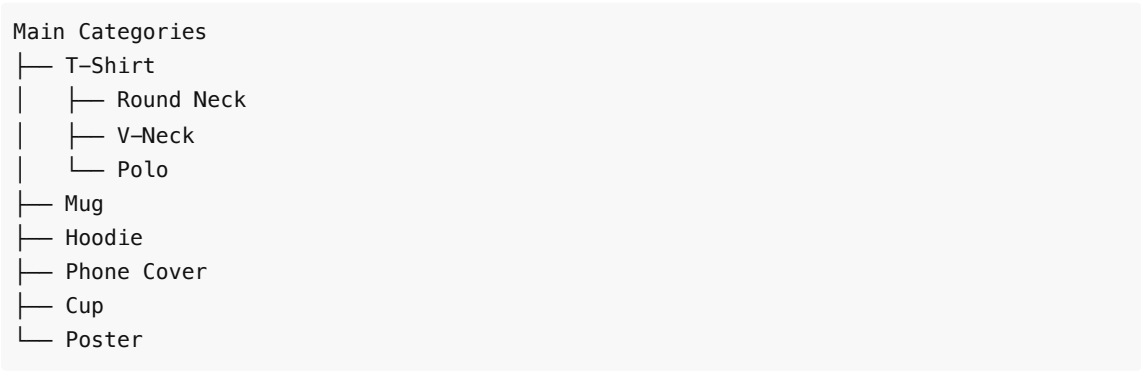

System Architecture

Core Components

```
┌─────────────────┐
│     User        │
└─────────────────┘
         │
         │
         ├──────▶ Browse Products
```

```
        |        └──────▶ Product Catalog
        |
        ├──────▶ Customize Design
        |        ├──────▶ Upload Design
        |        ├──────▶ Validate Design
        |        └──────▶ Generate Mockup
        |
        ├──────▶ Add to Cart
        |        └──────▶ Cart Management
        |
        ├──────▶ Place Order
        |        ├──────▶ Price Calculation
        |        ├──────▶ Stock Reservation
        |        └──────▶ Order Creation
        |
        ├──────▶ Track Order
        |        └──────▶ Order Timeline
        |
        └──────▶ Review Product
                 └──────▶ Rating & Review
```

Data Flow

```
User → Browse Products → Select Product → Customize Design
   ↓
Add to Cart → Apply Coupon → Calculate Price → Place Order
   ↓
Stock Reservation → Order Confirmation → Processing
   ↓
Production → Shipping → Delivery → Review
```

Product Catalog System

Category Structure

The system supports a hierarchical category structure:

```
Main Categories
├── T-Shirt
│   ├── Round Neck
│   ├── V-Neck
│   └── Polo
├── Mug
├── Hoodie
├── Phone Cover
├── Cup
└── Poster
```

Product Information

Each product includes:

- **Basic Info**: Name, description, images, thumbnail
- **Pricing**: Price, discount price, size-based pricing
- **Variants**: Colors, sizes with availability
- **Specifications**: Materials, product type, care instructions
- **Delivery**: Express/standard delivery options
- **Cashback**: Cashback percentage
- **Stock**: Real-time stock availability
- **Ratings**: Average rating and review count

Product Variants

Colors:

```
{
  "id": "1",
  "colorCode": "#000000",
  "colorName": "Black",
  "isAvailable": true
}
```

Sizes:

```
{
  "size": "M",
  "isAvailable": true,
  "price": 499
}
```

Product Search & Filters

Users can search and filter products by:

- Category and subcategory
- Price range (minPrice, maxPrice)
- Colors
- Sizes
- Search keywords
- Best sellers
- Availability

Shopping Cart Management

Cart Structure

Each cart item contains:

- **Product**: Product ID and details
- **Quantity**: Number of items
- **Variants**: Selected color and size
- **Design**: Custom design ID (optional)
- **Price**: Item price with discounts

- **Subtotal**: Quantity × Price

Cart Operations

1. **Add to Cart**

   - Validate product availability
   - Check stock for selected variant
   - Add item or update quantity if exists

2. **Update Cart Item**

   - Change quantity
   - Update color/size
   - Update design

3. **Remove Cart Item**

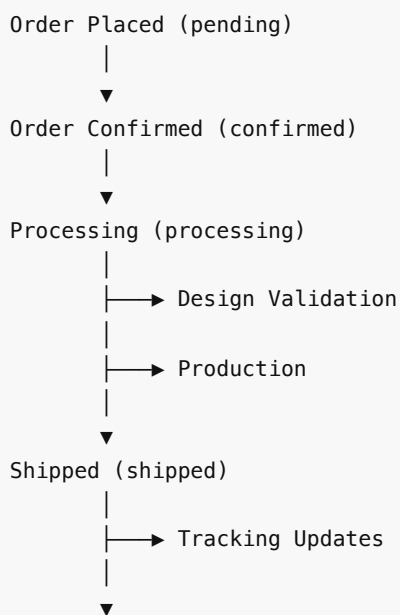   - Remove single item from cart

4. **Clear Cart**

   - Remove all items from cart
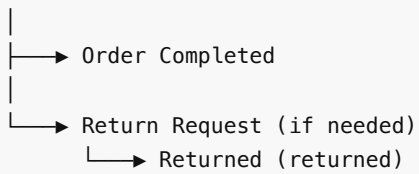
Cart Calculation

```
Subtotal = Sum of (Quantity × Price) for all items
Discount = Coupon discount (if applied)
Delivery Charges = Based on pincode and delivery type
Cashback = Sum of (Item Price × Cashback %) for all items
Total = Subtotal – Discount + Delivery Charges
```

Order Management

Order Lifecycle

```
Order Placed (pending)
      |
      ▼
Order Confirmed (confirmed)
      |
      ▼
Processing (processing)
      |
      ├─────▶ Design Validation
      |
      ├─────▶ Production
      |
      ▼
Shipped (shipped)
      |
      ├─────▶ Tracking Updates
      |
      ▼
```

```
Delivered (delivered)
       |
       ├────▶ Order Completed
       |
       └────▶ Return Request (if needed)
               └────▶ Returned (returned)
```

Order Statuses

- **pending**: Order placed, awaiting confirmation
- **confirmed**: Order confirmed, payment received
- **processing**: Order in production
- **shipped**: Order shipped, in transit
- **delivered**: Order delivered successfully
- **cancelled**: Order cancelled by user/admin
- **returned**: Order returned by user

Order Fields

Shipping Address:

- Name
- Phone
- Address
- City
- State
- Pincode
- Landmark

Order Items:

- Product details
- Quantity
- Color and size
- Design (if custom)
- Price per item
- Subtotal

Order Summary:

- Subtotal
- Discount (coupon)
- Delivery charges
- Cashback amount
- Total amount
- Payment method

Order Timeline

Each order maintains a timeline with automatic status updates:

```
[
  {
    "title": "Order Placed",
    "time": "10:30 AM",
```

```
    "status": "completed",
    "icon": "check_circle",
    "timestamp": "2024-01-01T10:30:00.000Z",
    "description": "Your order has been placed successfully"
  },
  {
    "title": "Order Confirmed",
    "time": "10:35 AM",
    "status": "completed",
    "icon": "check_circle",
    "timestamp": "2024-01-01T10:35:00.000Z",
    "description": "Payment received, order confirmed"
  },
  {
    "title": "Processing",
    "time": "Current",
    "status": "current",
    "icon": "hourglass_empty",
    "timestamp": "2024-01-01T11:00:00.000Z",
    "description": "Your order is being processed"
  }
]
```

Design & Customization

Design Upload

Users can upload custom designs for products:

- **File Types**: JPG, JPEG, PNG, PDF
- **Max Size**: 10MB per file
- **Validation**: Design validation for product compatibility
- **Storage**: Secure file storage

Design Validation

Before placing order, designs are validated:

- File format check
- Resolution check
- Color compatibility
- Print area validation
- File size validation

Mockup Generation

Generate product mockups with custom designs:

- Apply design to product
- Show design on selected color
- Preview in selected size
- Download mockup image

Design Workflow

```
Upload Design → Validate Design → Generate Mockup → Add to Cart → Place Order
```

Delivery System

Delivery Types

1. **Standard Delivery**

   - Default delivery option
   - 5-7 business days
   - Free or low-cost delivery
   - Standard delivery slots

2. **Express Delivery**

   - Fast delivery option
   - 2-3 business days
   - Additional express charges
   - Express delivery slots
   - Available for eligible products

Delivery Charges

Delivery charges are calculated based on:

- **Pincode**: Different rates for different areas
- **Delivery Type**: Standard or express
- **Order Value**: Free delivery above threshold
- **Product Weight**: Weight-based charges

Delivery Slots

Users can select delivery slots:

- **Standard Slots**: Available in 5-7 days
- **Express Slots**: Available in 2-3 days
- **Time Slots**: Morning, Afternoon, Evening
- **Date Selection**: Choose preferred delivery date

Address Suggestions

System provides address suggestions based on:

- Pincode
- City
- State
- Auto-complete functionality

Review & Rating System

Review Structure

Each review includes:

- **Rating**: 1-5 stars
- **Comment**: Review text
- **Order ID**: Associated order (optional)
- **User Info**: Reviewer details
- **Timestamp**: Review date

Product Ratings

Products display:

- **Average Rating**: Calculated from all reviews
- **Rating Distribution**: Count of each rating (1-5)
- **Total Reviews**: Total number of reviews
- **Recent Reviews**: Latest reviews with pagination

Review Workflow

```
Order Delivered → User can Review → Submit Review → Review Published
```

Coupon & Discount System

Coupon Types

1. **Percentage Discount**: 10% off, 20% off, etc.
2. **Fixed Amount**: ₹100 off, ₹500 off, etc.
3. **Free Delivery**: Free delivery coupon
4. **Minimum Purchase**: Discount on minimum purchase amount

Coupon Validation

Coupons are validated for:

- **Validity**: Start and end date
- **Usage Limit**: Maximum uses per user
- **Minimum Purchase**: Minimum order value
- **Applicable Products**: Product/category restrictions
- **Active Status**: Coupon must be active

Coupon Application

```
Enter Coupon Code → Validate Coupon → Apply Discount → Update Order Total
```

Data Models

PODCategory Model

```
{
  id: string;
  cat_name: string;
  cat_img: string;
```

```
    status: number;  // 1 = active, 0 = inactive
    parentCategoryId?: ObjectId;  // For subcategories
    order: number;  // Display order
    createdAt: Date;
    updatedAt: Date;
}
```

PODProduct Model

```
{
  id: string;
  productId: string;  // Display ID like "PROD000001"
  name: string;
  categoryId: ObjectId;
  subcategoryId?: ObjectId;
  imageUrls: string[];
  thumbnail: string;
  price: number;
  discountPrice: number;
  description: string;
  shortDescription: string;
  colors: Array<{
    id: string;
    colorCode: string;
    colorName: string;
    isAvailable: boolean;
  }>;
  sizes: Array<{
    size: string;
    isAvailable: boolean;
    price?: number;
  }>;
  materials?: string[];
  productType?: string;
  ratings: number[];  // Array of individual ratings
  averageRating: number;
  reviewCount: number;
  isExpressDelivery: boolean;
  expressDeliveryDays: number;
  standardDeliveryDays: number;
  expressDeliveryPrice: number;
  standardDeliveryPrice: number;
  cashbackPercentage: number;
  isActive: boolean;
  stock: {
    inStock: boolean;
    quantity: number;
  };
  specifications?: Record<string, any>;
  createdAt: Date;
```

```
    updatedAt: Date;
}
```

PODCart Model

```
{
  id: string;
  userId: ObjectId;
  items: Array<{
    productId: ObjectId;
    quantity: number;
    colorId: string;
    size: string;
    designId?: ObjectId;
    price: number;
    subtotal: number;
  }>;
  couponCode?: string;
  discountAmount: number;
  subtotal: number;
  total: number;
  createdAt: Date;
  updatedAt: Date;
}
```

PODOrder Model

```
{
  id: string;
  orderId: string;  // Display ID like "ORD000001"
  userId: ObjectId;
  items: Array<{
    productId: ObjectId;
    productName: string;
    quantity: number;
    colorId: string;
    colorName: string;
    size: string;
    designId?: ObjectId;
    designUrl?: string;
    price: number;
    subtotal: number;
  }>;
  shippingAddress: {
    name: string;
    phone: string;
    address: string;
    city: string;
    state: string;
```

```
    pincode: string;
    landmark?: string;
  };
  paymentMethod: 'upi' | 'card' | 'cod' | 'wallet';
  paymentStatus: 'pending' | 'completed' | 'failed' | 'refunded';
  orderStatus: 'pending' | 'confirmed' | 'processing' | 'shipped' |
              'delivered' | 'cancelled' | 'returned';
  subtotal: number;
  discountAmount: number;
  deliveryCharges: number;
  cashbackAmount: number;
  totalAmount: number;
  couponCode?: string;
  deliveryType: 'standard' | 'express';
  deliverySlot?: {
    date: string;
    timeSlot: string;
  };
  trackingNumber?: string;
  timeline: Array<{
    title: string;
    time: string;
    status: 'completed' | 'current' | 'pending';
    icon: string;
    timestamp: Date;
    description?: string;
  }>;
  cancelledAt?: Date;
  cancelledBy?: 'user' | 'admin';
  cancellationReason?: string;
  returnedAt?: Date;
  returnReason?: string;
  deliveredAt?: Date;
  createdAt: Date;
  updatedAt: Date;
}
```

PODDesign Model

```
{
  id: string;
  userId: ObjectId;
  productId: ObjectId;
  designName?: string;
  designUrl: string;
  fileType: string;
  fileSize: number;
  validationStatus: 'pending' | 'approved' | 'rejected';
  validationMessage?: string;
  mockupUrl?: string;
  isActive: boolean;
```

```
    createdAt: Date;
    updatedAt: Date;
  }
```

PODReview Model

```
{
  id: string;
  userId: ObjectId;
  productId: ObjectId;
  orderId?: ObjectId;
  rating: number;  // 1-5
  comment: string;
  isVerified: boolean;  // Verified purchase
  helpfulCount: number;
  createdAt: Date;
  updatedAt: Date;
}
```

PODCoupon Model

```
{
  id: string;
  couponCode: string;
  discountType: 'percentage' | 'fixed';
  discountValue: number;
  minPurchaseAmount?: number;
  maxDiscountAmount?: number;
  startDate: Date;
  endDate: Date;
  usageLimit?: number;  // Total uses
  usageLimitPerUser?: number;  // Uses per user
  applicableProducts?: ObjectId[];  // Specific products
  applicableCategories?: ObjectId[];  // Specific categories
  isActive: boolean;
  usedCount: number;
  createdAt: Date;
  updatedAt: Date;
}
```

PODBanner Model

```
{
  id: string;
  banner?: string;  // For carousel type
  title?: string;  // For promotional type
  subtitle?: string;
  imageUrl?: string;
```

```
    backgroundColor?: string;
    link: string;
    type: 'carousel' | 'promotional';
    order: number;
    isActive: boolean;
    createdAt: Date;
    updatedAt: Date;
}
```

API Endpoints

Catalog APIs

1. Get Categories

GET `/api/v1/pod/categories`

Get all active product categories.

Response:

```
{
  "success": true,
  "data": {
    "categories": [
      {
        "id": "...",
        "cat_name": "T-Shirt",
        "cat_img": "https://...",
        "status": 1,
        "order": 1
      }
    ]
  }
}
```

2. Get Subcategories

GET `/api/v1/pod/categories/:categoryId/subcategories`

Get subcategories for a specific category.

Query Params:

- `categoryId` : Category ID

Response:

```
{
  "success": true,
  "data": {
    "subcategories": [
      {
```

```
        "id": "...",
        "cat_name": "Round Neck",
        "cat_img": "https://...",
        "parentCategoryId": "...",
        "order": 1
      }
    ]
  }
}
```

3. Get Products

GET `/api/v1/pod/products`

Get products with filters and pagination.

Query Params:

- `categoryId` : Filter by category (optional)
- `subcategoryId` : Filter by subcategory (optional)
- `page` : Page number (default: 1)
- `limit` : Items per page (default: 20)
- `search` : Search keyword (optional)
- `minPrice` : Minimum price (optional)
- `maxPrice` : Maximum price (optional)
- `colors` : Color IDs array (optional)
- `sizes` : Size array (optional)

Response:

```
{
  "success": true,
  "data": {
    "products": [...],
    "pagination": {
      "page": 1,
      "limit": 20,
      "total": 100,
      "pages": 5
    }
  }
}
```

4. Get Product Details

GET `/api/v1/pod/products/:productId`

Get detailed information about a specific product.

Response:

```
{
  "success": true,
```

```
    "data": {
      "product": {
        "id": "...",
        "productId": "PROD000001",
        "name": "Classic Black T-Shirt",
        "description": "...",
        "price": 599,
        "discountPrice": 499,
        "colors": [...],
        "sizes": [...],
        "stock": {
          "inStock": true,
          "quantity": 150
        },
        "averageRating": 4.72,
        "reviewCount": 125
      }
    }
  }
```

5. Search Products

GET `/api/v1/pod/products/search`

Search products by keyword.

Query Params:

- `q` : Search query
- `page` : Page number
- `limit` : Items per page

6. Get Best Sellers

GET `/api/v1/pod/products/best-sellers`

Get best-selling products.

Query Params:

- `limit` : Number of products (default: 10)

7. Get Banners

GET `/api/v1/pod/banners`

Get promotional banners.

Query Params:

- `type` : `carousel | promotional` (optional)

Cart APIs

8. Add to Cart

POST `/api/v1/pod/cart/add`

Add product to cart.

Request Body:

```json
{
  "productId": "PROD000001",
  "quantity": 2,
  "colorId": "1",
  "size": "M",
  "designId": "design_id"  // Optional
}
```

Response:

```json
{
  "success": true,
  "data": {
    "cart": {
      "items": [...],
      "subtotal": 998,
      "total": 998
    }
  }
}
```

9. Get Cart Items

GET `/api/v1/pod/cart`

Get all items in user's cart.

Response:

```json
{
  "success": true,
  "data": {
    "cart": {
      "items": [
        {
          "id": "...",
          "product": {...},
          "quantity": 2,
          "colorId": "1",
          "size": "M",
          "price": 499,
          "subtotal": 998
        }
      ],
      "subtotal": 998,
      "discountAmount": 0,
      "total": 998
    }
```

```
      }
  }
```

10. Update Cart Item

PATCH `/api/v1/pod/cart/:cartItemId`

Update cart item quantity or variants.

Request Body:

```
{
    "quantity": 3,
    "colorId": "2",
    "size": "L"
}
```

11. Remove Cart Item

DELETE `/api/v1/pod/cart/:cartItemId`

Remove item from cart.

12. Clear Cart

DELETE `/api/v1/pod/cart`

Remove all items from cart.

Order APIs

13. Place Order

POST `/api/v1/pod/orders`

Place a new order.

Request Body:

```
{
  "items": [
    {
      "productId": "PROD000001",
      "quantity": 2,
      "colorId": "1",
      "size": "M",
      "designId": "design_id"
    }
  ],
  "shippingAddress": {
    "name": "John Doe",
    "phone": "9876543210",
    "address": "123 Main St",
    "city": "Mumbai",
```

```
      "state": "Maharashtra",
      "pincode": "400001",
      "landmark": "Near Park"
    },
    "paymentMethod": "upi",
    "couponCode": "SAVE10",  // Optional
    "deliveryType": "standard",
    "deliverySlot": {  // Optional
      "date": "2024-01-15",
      "timeSlot": "10:00 AM - 2:00 PM"
    }
  }
```

Response:

```
{
  "success": true,
  "data": {
    "order": {
      "orderId": "ORD000001",
      "items": [...],
      "subtotal": 998,
      "discountAmount": 99.8,
      "deliveryCharges": 0,
      "cashbackAmount": 99.8,
      "totalAmount": 898.2,
      "orderStatus": "pending",
      "timeline": [...]
    }
  }
}
```

14. Get Order Details

GET `/api/v1/pod/orders/:orderId`

Get detailed information about an order.

Response:

```
{
  "success": true,
  "data": {
    "order": {
      "orderId": "ORD000001",
      "items": [...],
      "shippingAddress": {...},
      "orderStatus": "processing",
      "timeline": [...],
      "trackingNumber": "TRACK123456"
    }
```

```
    }
  }
```

15. Track Order

GET `/api/v1/pod/orders/track/:orderId`

Track order status (public endpoint, no auth required).

Response:

```
{
  "success": true,
  "data": {
    "orderId": "ORD000001",
    "orderStatus": "shipped",
    "trackingNumber": "TRACK123456",
    "currentLocation": "Mumbai Warehouse",
    "estimatedDelivery": "2024-01-15",
    "timeline": [...]
  }
}
```

16. Get User Orders

GET `/api/v1/pod/orders`

Get all orders for the authenticated user.

Query Params:

- `page` : Page number
- `limit` : Items per page
- `status` : Filter by status (optional)

Response:

```
{
  "success": true,
  "data": {
    "orders": [...],
    "pagination": {
      "page": 1,
      "limit": 10,
      "total": 25,
      "pages": 3
    }
  }
}
```

17. Cancel Order

POST `/api/v1/pod/orders/:orderId/cancel`

Cancel an order.

Request Body:

```json
{
  "reason": "Changed my mind"
}
```

Response:

```json
{
  "success": true,
  "data": {
    "order": {
      "orderId": "ORD000001",
      "orderStatus": "cancelled",
      "cancelledAt": "2024-01-02T10:00:00.000Z"
    }
  }
}
```

18. Return Order

POST `/api/v1/pod/orders/:orderId/return`

Request order return.

Request Body:

```json
{
  "reason": "Defective product",
  "description": "Product received was damaged"
}
```

Design APIs

19. Upload Design

POST `/api/v1/pod/designs/upload`

Upload custom design file.

Form Data:

- `file` : File (multipart/form-data)
- `productId` : Product ID
- `designName` : Design name (optional)

Response:

```json
{
  "success": true,
  "data": {
```

```
    "design": {
      "id": "...",
      "designUrl": "https://...",
      "validationStatus": "pending"
    }
  }
}
```

20. Validate Design

POST `/api/v1/pod/designs/validate`

Validate design for product compatibility.

Request Body:

```
{
  "designId": "design_id",
  "productId": "PROD000001",
  "colorId": "1",
  "size": "M"
}
```

Response:

```
{
  "success": true,
  "data": {
    "isValid": true,
    "validationMessage": "Design is valid for this product",
    "warnings": []
  }
}
```

21. Generate Mockup

POST `/api/v1/pod/mockup/generate`

Generate product mockup with design.

Request Body:

```
{
  "designId": "design_id",
  "productId": "PROD000001",
  "colorId": "1",
  "size": "M"
}
```

Response:

```
{
  "success": true,
  "data": {
    "mockupUrl": "https://...",
    "design": {
      "id": "...",
      "designUrl": "https://..."
    }
  }
}
```

Review APIs

22. Get Product Reviews

GET `/api/v1/pod/products/:productId/reviews`

Get reviews for a product.

Query Params:

- `page` : Page number
- `limit` : Items per page
- `rating` : Filter by rating (1-5, optional)

Response:

```
{
  "success": true,
  "data": {
    "reviews": [
      {
        "id": "...",
        "user": {
          "name": "John Doe"
        },
        "rating": 5,
        "comment": "Great product!",
        "isVerified": true,
        "createdAt": "2024-01-01T10:00:00.000Z"
      }
    ],
    "pagination": {...},
    "summary": {
      "averageRating": 4.72,
      "totalReviews": 125,
      "ratingDistribution": {
        "5": 80,
        "4": 30,
        "3": 10,
        "2": 3,
        "1": 2
```

```
        }
      }
    }
  }
```

23. Add Review

POST `/api/v1/pod/products/:productId/reviews`

Add a review for a product.

Request Body:

```
{
  "rating": 5,
  "comment": "Great product!",
  "orderId": "order_id"  // Optional, for verified purchase
}
```

Response:

```
{
  "success": true,
  "data": {
    "review": {
      "id": "...",
      "rating": 5,
      "comment": "Great product!",
      "isVerified": true
    }
  }
}
```

Delivery APIs

24. Get Express Delivery Info

GET `/api/v1/pod/express-delivery/info`

Get information about express delivery.

Response:

```
{
  "success": true,
  "data": {
    "isAvailable": true,
    "deliveryDays": 2,
    "charges": 99,
    "description": "Get your order delivered in 2-3 business days"
```

```
      }
  }
```

25. Check Express Delivery

GET `/api/v1/pod/products/:productId/express-delivery`

Check if express delivery is available for a product.

Query Params:

- `pincode` : Delivery pincode

Response:

```json
{
  "success": true,
  "data": {
    "isAvailable": true,
    "deliveryDays": 2,
    "charges": 99,
    "estimatedDelivery": "2024-01-05"
  }
}
```

26. Get Delivery Charges

GET `/api/v1/pod/delivery/charges`

Calculate delivery charges.

Query Params:

- `pincode` : Delivery pincode
- `items` : JSON array of items with productId and quantity
- `isExpress` : Boolean (true/false)

Response:

```json
{
  "success": true,
  "data": {
    "deliveryCharges": 0,
    "deliveryType": "standard",
    "estimatedDays": 5,
    "freeDeliveryThreshold": 999,
    "isFreeDelivery": true
  }
}
```

27. Get Address Suggestions

GET `/api/v1/pod/addresses/suggestions`

Get address suggestions based on pincode.

Query Params:

- `pincode` : Pincode to search

Response:

```json
{
  "success": true,
  "data": {
    "suggestions": [
      {
        "address": "123 Main St, Mumbai",
        "city": "Mumbai",
        "state": "Maharashtra",
        "pincode": "400001"
      }
    ]
  }
}
```

Additional APIs

28. Apply Coupon

POST `/api/v1/pod/coupons/apply`

Apply discount coupon to cart/order.

Request Body:

```json
{
  "couponCode": "SAVE10",
  "items": [
    {
      "productId": "PROD000001",
      "quantity": 2,
      "price": 499
    }
  ]
}
```

Response:

```json
{
  "success": true,
  "data": {
    "coupon": {
      "code": "SAVE10",
      "discountType": "percentage",
      "discountValue": 10
    },
    "discountAmount": 99.8,
```

```
      "newTotal": 898.2
    }
  }
}
```

29. Get Packaging Categories

GET `/api/v1/pod/packaging/categories`

Get packaging categories (for gift packaging options).

Response:

```
{
  "success": true,
  "data": {
    "categories": [
      {
        "id": "...",
        "name": "Gift Box",
        "image": "https://...",
        "price": 50
      }
    ]
  }
}
```

30. Get Packaging Products

GET `/api/v1/pod/packaging/products`

Get packaging products for a category.

Query Params:

- `categoryId` : Packaging category ID
- `page` : Page number
- `limit` : Items per page

Workflow Examples

Complete Order Flow

```
1. User browses products
   GET /api/v1/pod/products?categoryId=...

2. User views product details
   GET /api/v1/pod/products/PROD000001

3. User uploads custom design (optional)
   POST /api/v1/pod/designs/upload
   POST /api/v1/pod/designs/validate
   POST /api/v1/pod/mockup/generate
```

```
4. User adds product to cart
   POST /api/v1/pod/cart/add

5. User views cart
   GET /api/v1/pod/cart

6. User applies coupon (optional)
   POST /api/v1/pod/coupons/apply

7. User checks delivery charges
   GET /api/v1/pod/delivery/charges?pincode=400001

8. User places order
   POST /api/v1/pod/orders

9. User tracks order
   GET /api/v1/pod/orders/track/ORD000001

10. Order delivered, user reviews
    POST /api/v1/pod/products/PROD000001/reviews
```

Design Customization Flow

```
1. User selects product
   GET /api/v1/pod/products/PROD000001

2. User uploads design
   POST /api/v1/pod/designs/upload
   {
     "file": <file>,
     "productId": "PROD000001"
   }

3. System validates design
   POST /api/v1/pod/designs/validate
   {
     "designId": "...",
     "productId": "PROD000001",
     "colorId": "1",
     "size": "M"
   }

4. User generates mockup
   POST /api/v1/pod/mockup/generate
   {
     "designId": "...",
     "productId": "PROD000001",
     "colorId": "1",
     "size": "M"
   }
```

```
5. User adds to cart with design
   POST /api/v1/pod/cart/add
   {
     "productId": "PROD000001",
     "quantity": 1,
     "colorId": "1",
     "size": "M",
     "designId": "..."
   }
```

Pricing Calculation

Price Components

```
Base Price = Product Price (or size-specific price)
Quantity = Number of items
Subtotal = Base Price × Quantity

Discount = Coupon Discount (if applied)
  - Percentage: (Subtotal × Discount%) / 100
  - Fixed: Discount Amount
  - Max Discount: Applied if discount exceeds max limit

Delivery Charges = Based on:
  - Pincode
  - Delivery Type (standard/express)
  - Order Value (free delivery threshold)
  - Product Weight

Cashback = Sum of (Item Price × Cashback%) for all items

Total = Subtotal - Discount + Delivery Charges
```

Example Calculation

```
Product: T-Shirt
Price: ₹499
Quantity: 2
Subtotal: ₹998

Coupon: SAVE10 (10% off)
Discount: ₹99.8

Delivery: Standard (Free above ₹999)
Delivery Charges: ₹0 (Subtotal - Discount = ₹898.2, below threshold)
Actual Delivery Charges: ₹50

Cashback: 10% of ₹998 = ₹99.8
```

```
Total = ₹998 − ₹99.8 + ₹50 = ₹948.2
Cashback Earned = ₹99.8
```

Stock Management

Stock Checking

Before adding to cart or placing order:

1. Check product stock availability
2. Check variant (color/size) availability
3. Reserve stock when order is placed
4. Release stock if order is cancelled

Stock Reservation

```
Order Placed → Stock Reserved → Order Confirmed → Stock Deducted
Order Cancelled → Stock Released
```

Stock Updates

- Real-time stock checking
- Automatic stock deduction on order confirmation
- Stock release on order cancellation
- Low stock alerts (future feature)

Order Timeline

Automatic Timeline Updates

The system automatically updates order timeline based on status changes:

Order Placed:

```
{
  "title": "Order Placed",
  "status": "completed",
  "icon": "check_circle",
  "timestamp": "2024-01-01T10:30:00.000Z"
}
```

Order Confirmed (after payment):

```
{
  "title": "Order Confirmed",
  "status": "completed",
  "icon": "check_circle",
  "timestamp": "2024-01-01T10:35:00.000Z"
}
```

Processing:

```json
{
  "title": "Processing",
  "status": "current",
  "icon": "hourglass_empty",
  "timestamp": "2024-01-01T11:00:00.000Z"
}
```

Shipped:

```json
{
  "title": "Shipped",
  "status": "completed",
  "icon": "local_shipping",
  "timestamp": "2024-01-03T10:00:00.000Z",
  "trackingNumber": "TRACK123456"
}
```

Delivered:

```json
{
  "title": "Delivered",
  "status": "completed",
  "icon": "check_circle",
  "timestamp": "2024-01-05T14:00:00.000Z"
}
```

Configuration

Environment Variables

```
# POD Configuration
POD_EXPRESS_DELIVERY_ENABLED=true
POD_EXPRESS_DELIVERY_DAYS=2
POD_EXPRESS_DELIVERY_CHARGES=99
POD_STANDARD_DELIVERY_DAYS=5
POD_FREE_DELIVERY_THRESHOLD=999
POD_MAX_DESIGN_FILE_SIZE=10485760  # 10MB
POD_ALLOWED_DESIGN_TYPES=jpg,jpeg,png,pdf
```

Default Settings

- **Standard Delivery**: 5-7 business days
- **Express Delivery**: 2-3 business days
- **Free Delivery Threshold**: ₹999
- **Max Design File Size**: 10MB
- **Allowed Design Types**: JPG, JPEG, PNG, PDF
- **Cashback Percentage**: Product-specific (0-20%)

Database Indexes

```
// Product indexes
{ categoryId: 1, isActive: 1 }
{ subcategoryId: 1, isActive: 1 }
{ productId: 1 }
{ name: 'text', description: 'text' }  // Text search

// Order indexes
{ userId: 1, orderStatus: 1 }
{ orderId: 1 }
{ trackingNumber: 1 }
{ createdAt: -1 }

// Cart indexes
{ userId: 1 }

// Review indexes
{ productId: 1, createdAt: -1 }
{ userId: 1, productId: 1 }  // One review per user per product

// Design indexes
{ userId: 1 }
{ productId: 1, validationStatus: 1 }
```

Testing Checklist

API Testing

- ☐ Product catalog endpoints
- ☐ Cart management
- ☐ Order placement
- ☐ Order tracking
- ☐ Design upload and validation
- ☐ Mockup generation
- ☐ Review submission
- ☐ Coupon application
- ☐ Delivery charge calculation
- ☐ Stock management
- ☐ Error handling
- ☐ Authentication
- ☐ Validation
- ☐ Pagination

Integration Testing

- ☐ Complete order flow

- ☐ Design customization flow
- ☐ Cart to order conversion
- ☐ Stock reservation and release
- ☐ Order cancellation
- ☐ Order return
- ☐ Timeline updates
- ☐ Cashback calculation
- ☐ Coupon validation

Troubleshooting

Common Issues

1. **Product Not Found**

   - Verify productId format
   - Check product is active
   - Ensure product exists

2. **Stock Not Available**

   - Check product stock
   - Verify variant (color/size) availability
   - Check if stock was reserved

3. **Design Upload Fails**

   - Check file size (max 10MB)
   - Verify file type (JPG, PNG, PDF only)
   - Check uploads directory permissions

4. **Order Placement Fails**

   - Verify cart items
   - Check stock availability
   - Validate shipping address
   - Verify payment method

5. **Coupon Not Applied**

   - Check coupon validity dates
   - Verify minimum purchase amount
   - Check usage limits
   - Ensure coupon is active

6. **Delivery Charges Incorrect**

   - Verify pincode
   - Check delivery type
   - Verify free delivery threshold
   - Check product weight

Summary

The POD (Print On Demand) System provides a complete solution for:

- ✅ Product catalog browsing with filters
- ✅ Shopping cart management
- ✅ Order placement and tracking
- ✅ Custom design upload and validation
- ✅ Product mockup generation
- ✅ Delivery management (standard & express)
- ✅ Review and rating system
- ✅ Coupon and discount system
- ✅ Cashback rewards
- ✅ Real-time stock management
- ✅ Automatic order timeline updates
- ✅ Order cancellation and returns

All processes are automated, secure, and designed to provide excellent user experience while maintaining efficiency for order processing and fulfillment.