const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

name: { type: String, required: true },

email: { type: String, unique: true, required: true },

password: { type: String, required: true },

role: { type: String, enum: ['customer', 'admin'], default: 'customer' },

addres:[]

createdAt: { type: Date, default: Date.now }

});

const User = mongoose.model('User', userSchema);

module.exports = User;

const mongoose = require('mongoose');

const cartSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

products: [

{

productId: { type: mongoose.Schema.Types.ObjectId, ref: 'Product', required: true },

quantity: { type: Number, required: true }

}

],

createdAt: { type: Date, default: Date.now },

updatedAt: { type: Date, default: Date.now }

});

const Cart = mongoose.model('Cart', cartSchema);

module.exports = Cart;

----------------------------------------------------------------------------------------------------------------------

const mongoose = require('mongoose');

const wishlistSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

products: [{ type: mongoose.Schema.Types.ObjectId, ref: 'Product' }],

createdAt: { type: Date, default: Date.now },

updatedAt: { type: Date, default: Date.now }

});

const Wishlist = mongoose.model('Wishlist', wishlistSchema);

module.exports = Wishlist;

----------------------------------------------------------------------------------------------------------------------

const mongoose = require('mongoose');

const orderSchema = new mongoose.Schema({

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

products: [

{

productId: { type: mongoose.Schema.Types.ObjectId, ref: 'Product', required: true },

quantity: { type: Number, required: true }

}

],

totalPrice: { type: Number, required: true },

status: {

type: String,

enum: ['pending', 'shipped', 'delivered', 'cancelled'],

default: 'pending'

},

paymentMethod: { type: String, required: true },

shippingAddress: {

addressLine1: { type: String, required: true },

addressLine2: { type: String },

city: { type: String, required: true },

state: { type: String, required: true },

country: { type: String, required: true },

zipCode: { type: String, required: true }

},

createdAt: { type: Date, default: Date.now },

updatedAt: { type: Date, default: Date.now }

});

const Order = mongoose.model('Order', orderSchema);

module.exports = Order;

----------------------------------------------------------------------------------------------------------------------

const userWithCart = await User.findById(userId).populate('cart');

This would fetch a user along with their associated cart data.

Populating the cart or wishlist for a user

const userWithCart = await User.findById(userId).populate({

path: 'cart',

populate: { path: 'products.productId', model: 'Product' }

});

for orders:

const userWithOrders = await User.findById(userId).populate({

path: 'orders',

populate: { path: 'products.productId', model: 'Product' }

});