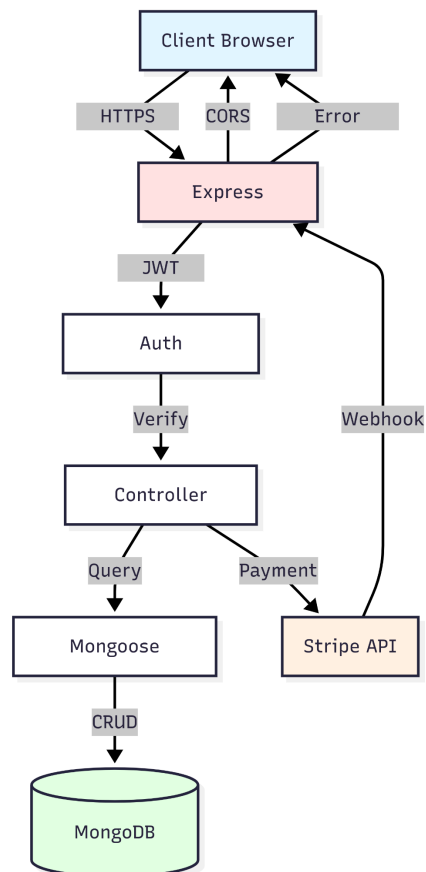


E-Commerce Platform - Design Documentation

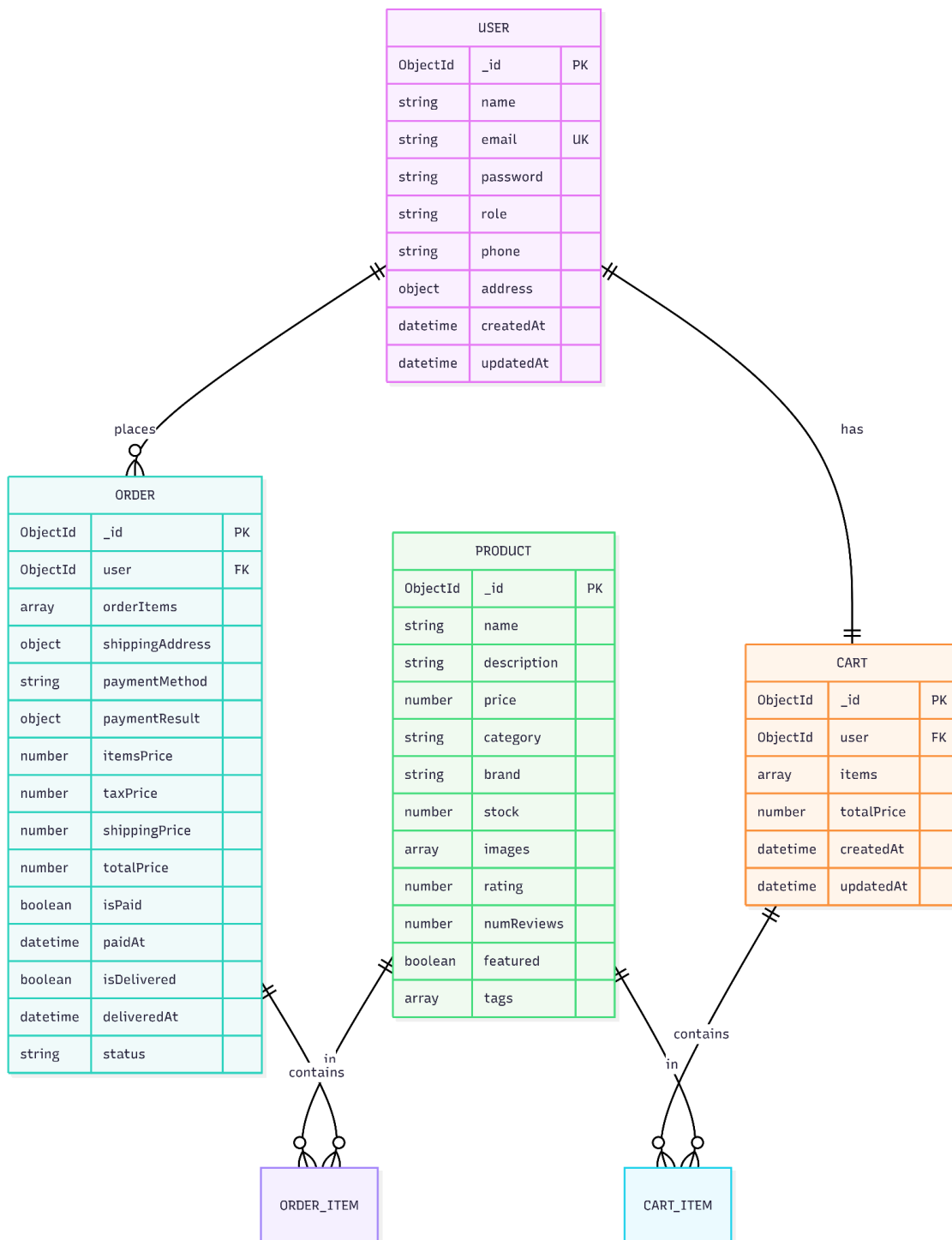
1. System Architecture

1.1 High-Level Architecture



2. Database Design

2.1 Entity Relationship Diagram (ERD)



Indexes:

- User.email (unique)
 - Product (text index on: name, description, tags)
 - Product.category
 - Product.featured
 - Cart.user (unique)
 - Order.user
 - Order.status
-

3. API Design

3.1 API Endpoint Summary

Method	Endpoint	Auth	Admin	Description
POST	/auth/register	✗	✗	Register new user
POST	/auth/login	✗	✗	Login user
GET	/auth/me	✓	✗	Get current user
GET	/products	✗	✗	Get all products
GET	/products/:id	✗	✗	Get single product
POST	/products	✓	✓	Create product
PUT	/products/:id	✓	✓	Update product
DELETE	/products/:id	✓	✓	Delete product
GET	/cart	✓	✗	Get user cart
POST	/cart/add	✓	✗	Add to cart
PUT	/cart/update	✓	✗	Update cart item
DELETE	/cart/remove/:id	✓	✗	Remove from cart
POST	/orders	✓	✗	Create order
GET	/orders/myorders	✓	✗	Get user orders
GET	/orders/:id	✓	✗	Get order details
GET	/orders	✓	✓	Get all orders
PUT	/orders/:id/status	✓	✓	Update order status
POST	/payment/create-intent	✓	✗	Create payment
POST	/payment/confirm	✓	✗	Confirm payment

4. Summary

4.1 Architecture Highlights

- ✓ MVC Pattern - Separation of concerns
- ✓ RESTful API - Standard HTTP methods
- ✓ JWT Authentication - Stateless auth
- ✓ Role-Based Access - Admin & customer roles
- ✓ Mongoose ODM - Schema validation
- ✓ Stripe Integration - Secure payments
- ✓ Error Handling - Centralized middleware
- ✓ Input Validation - Multiple layers
- ✓ MongoDB Indexes - Optimized queries

4.2 Security Highlights

- ✓ Password Hashing - Bcrypt with salt
- ✓ JWT Tokens - Signed & expiring
- ✓ Input Sanitization - XSS/NoSQL injection prevention
- ✓ CORS Protection - Whitelist origins
- ✓ Webhook Verification - Stripe signatures
- ✓ HTTPS Ready - Secure transport
- ✓ Environment Variables - Secret management

4.3 Key Design Decisions

Why MongoDB?

- Flexible schema for e-commerce products
- Easy to scale horizontally
- Rich query capabilities
- Native JSON support

Why JWT?

- Stateless authentication
- Mobile-friendly
- No server sessions needed
- Easy to scale

Why Stripe?

- Industry standard
- PCI-DSS compliant
- Great developer experience
- Strong webhook support

Why Express?

- Minimal overhead
- Flexible middleware

- Large ecosystem
- Easy to learn