

Technical Documentation for Rental E-commerce Platform

1. System Architecture Document

Overview:

- Frontend: User-facing interface (React/Next.js for Rental E-commerce website).
- Backend: Handles business logic, API interactions (Node.js/Express.js).
- Database: Stores product details, customer data, orders, and payment information (MongoDB or PostgreSQL).
- Third-party APIs: Used for payments (Stripe/PayPal) and shipment tracking.
- Deployment: Hosted on cloud platforms like AWS/Heroku for backend and Netlify/Vercel for frontend.

Interactions:

User sends requests to the frontend -> Backend processes requests and queries database -> Responses sent back to the frontend.

2. API Specification Document

Endpoints:

- GET /products
 - Description: Retrieve all available products for rental.
 - Response: [{ productId, productName, productDetails, price, availability }]
- POST /orders
 - Description: Create a new rental order.
 - Payload: { customerId, productId, rentalDuration, paymentDetails }
 - Response: { orderId, status, totalAmount }
- GET /shipments
 - Description: Track the delivery/return status of a product.

4. Data Schema Design

Entities:

- Products: { productId, name, details, price, availability }
- Customers: { customerId, name, contactInfo }
- Orders: { orderId, customerId, productId, rentalDuration, paymentStatus }
- Shipments: { shipmentId, orderId, status, expectedDeliveryDate }

Relationships:

- One customer can have multiple orders.
- One product can be part of multiple orders but not at the same time (availability).

5. Technical Roadmap

Phase 1: Setup

- Set up the project environment for backend and frontend.
- Configure the database schema.
- Integrate necessary libraries for frontend and backend.

Phase 2: Core Features

- Develop APIs for products, orders, and shipments.
- Create the user login and product browsing interface.

Phase 3: Payment and Shipment

- Integrate third-party APIs (e.g., Stripe for payments, shipping APIs for tracking).
- Test payment and shipment workflows.

Phase 4: Testing and Deployment

- Perform end-to-end testing for user and admin workflows.
- Deploy the application to cloud platforms.



API SPECIFICATION DOCUMENT

