**Comprehensive Shipping Options Implementation Guide**

**Overview**

This guide details the implementation of three shipping methods: Self Shipping, Automatic Shipping, and Hyper Local Delivery, covering backend logic, database structures, and API interactions.

**1. Self Shipping**

**Concept**

* Vendors independently manage shipping
* Manually input shipping provider and tracking information

**Implementation Details**

**Database Schema**

sql

Copy

CREATE TABLE Orders (

OrderID INT PRIMARY KEY,

CustomerID INT,

VendorID INT,

ShippingMethod VARCHAR(50),

ProviderName VARCHAR(100),

TrackingID VARCHAR(100),

Status VARCHAR(20)

);

**API Endpoint**

**Endpoint:** POST /orders/{orderId}/self-shipping

**Request Payload:**

json

Copy

{

"providerName": "Example Courier",

"trackingId": "12345ABC"

}

**Response:**

json

Copy

{

"message": "Tracking details added successfully.",

"status": "Shipped"

}

**Workflow**

1. Order placed with Self Shipping method
2. Vendor inputs shipping details
3. System updates order status to "Shipped"

**2. Automatic Shipping**

**Concept**

* Integration with courier services (e.g., Shiprocket)
* Automated shipping and tracking

**Implementation Details**

**Database Schema**

sql

Copy

CREATE TABLE ShippingIntegration (

IntegrationID INT PRIMARY KEY,

ProviderName VARCHAR(100),

APIKey VARCHAR(255),

BaseURL VARCHAR(255)

);

*-- Updated Orders Table*

ALTER TABLE Orders ADD COLUMN ShippingCost DECIMAL(10,2);

**API Endpoint**

**Endpoint:** POST /orders/{orderId}/automatic-shipping

**Request Payload:**

json

Copy

{

"destination": {

"city": "Mumbai",

"pincode": "400001"

},

"products": [

{ "id": 1, "weight": 2.5, "dimensions": "10x5x3" }

]

}

**Response:**

json

Copy

{

"trackingId": "SHIP123456",

"shippingCost": 150.00

}

**Workflow**

1. Calculate shipping rates dynamically
2. Integrate with Shiprocket API
3. Generate tracking ID automatically
4. Process shipment

**3. Hyper Local Delivery**

**Concept**

* Same-day or next-day delivery within specific cities
* Uses local delivery providers (e.g., Borzo)

**Implementation Details**

**Database Schema**

sql

Copy

CREATE TABLE ServiceAreas (

AreaID INT PRIMARY KEY,

City VARCHAR(100),

CutoffTime TIME,

VendorID INT

);

*-- Updated Orders Table*

ALTER TABLE Orders ADD COLUMN DeliveryTime TIME;

**API Endpoint**

**Endpoint:** POST /orders/{orderId}/hyper-local

**Request Payload:**

json

Copy

{

"city": "Bengaluru",

"cutoffTime": "18:00:00"

}

**Response:**

json

Copy

{

"message": "Order scheduled for same-day delivery.",

"deliveryTime": "16:30:00"

}

**Workflow**

1. Verify delivery address is in serviceable city
2. Check order placement time against cutoff
3. Schedule same-day or next-day delivery
4. Integrate with local delivery provider API

**Key Considerations**

**Shipping Method Selection**

* **Self Shipping**: Vendor-managed, manual tracking
* **Automatic Shipping**: Admin-configured, API-integrated
* **Hyper Local Delivery**: Location and time-sensitive

**System Flexibility**

* Configurable at admin, vendor, and product levels
* Supports multiple shipping providers
* Dynamic rate and delivery time calculations

**Recommended Next Steps**

1. Implement robust API integration
2. Create comprehensive error handling
3. Develop vendor and admin management interfaces
4. Set up monitoring for shipping processes