App Diagram

User Side

- Authentication (Signup, Login, JWT)
- Browse Food Items
- Add to Cart & Checkout
- Order Tracking
- Payment Integration

Admin Side

- Manage Users
- Add/Edit/Delete Food Items
- Manage Orders
- View Reports & Analytics

1. Define the Main Entities (Collections in MongoDB)

In a food service app, the key collections (tables in SQL) are:

1. ***Users** → Stores customer & admin details*
2. ***Admin** → Stores admin login details*
3. ***Food Categories** → Stores different food categories*
4. ***Food Items** → Stores menu items with details*
5. ***Cart** → Stores items added to the cart*
6. ***Orders** → Stores order details*
7. ***Payments** → Stores payment details*
8. ***Reviews** → Stores customer feedback*
9. ***Delivery Address** → Stores user addresses*

---

2. Database Schema (MongoDB Collections)

User Schema (users)

- *_id (ObjectId) → Unique identifier*
- *name (String) → Full name*
- *email (String) → User email (Unique)*
- *password (String) → Hashed password*
- *role (String) → "user" or "admin"*
- *phone (String) → Contact number*
- *addresses (Array) → List of user addresses*
- *orders (Array of ObjectId) → References orders*
- *createdAt (MomntJs)*

Admin Schema (admins)

- *_id (ObjectId)*
- *email (String) → Unique admin email*
- *password (String) → Hashed password*
- *role (String) → "admin"*

Food Category Schema (categories)

- *_id (ObjectId)*
- *name (String) → Category name (e.g., Fast Food, Desserts)*
- *description (String)*

Food Item Schema (foodItems)

- *_id (ObjectId)*
- *name (String) → Food item name*
- *description (String)*
- *price (Number)*
- *categoryId (ObjectId) → References categories*
- *image (String) → Food image URL*
- *stock (Boolean) → Available or not*
- *rating (Number)*

Cart Schema (carts)

- *_id (ObjectId)*
- *userId (ObjectId) → References users*
- *items (Array) → List of food items {foodItemId, quantity}*
- *totalPrice (Number)*
- *updatedAt (MomntJs)*

Order Schema (orders)

- *_id (ObjectId)*
- *userId (ObjectId) → References users*
- *items (Array) → List of ordered items {foodItemId, quantity, price}*
- *totalAmount (Number)*
- *status (String) → "Pending", "Confirmed", "Out for Delivery", "Delivered"*
- *paymentId (ObjectId) → References payments*
- *createdAt (MomntJs)*

Payment Schema (payments)

- *_id (ObjectId)*
- *userId (ObjectId) → References users*

- *orderId (ObjectId) → References orders*
- *paymentMethod (String) → "Card", "UPI", "Cash on Delivery"*
- *status (String) → "Success" or "Failed"*
- *createdAt (Date)*

Review Schema (reviews)

- *_id (ObjectId)*
- *userId (ObjectId) → References users*
- *foodItemId (ObjectId) → References foodItems*
- *rating (Number) → 1-5*
- *comment (String)*
- *createdAt (Date)*

---

3. Steps to Build the Food Service App

Step 1: Setup Backend (Node.js + Express + MongoDB)

1. *Install dependencies: express, mongoose, bcryptjs, jsonwebtoken, cors, dotenv*
2. *Connect MongoDB (mongoose.connect)*
3. *Create models (User, Admin, FoodItem, Category, Cart, Order, Payment)*

Step 2: Implement Authentication

1. *User & Admin **Login/Register API***
2. *JWT token generation for authentication*
3. *Middleware to protect admin routes*

Step 3: Implement Admin Features

1. ***Add, Edit, Delete Food Items***
2. ***Manage Orders (Update status: Pending → Delivered)***
3. ***Manage Users (View all users, delete if necessary)***
4. ***View Payments***

Step 4: Implement User Features

1. ***User Registration & Login***
2. ***Browse Food Items by Category***
3. ***Add to Cart & View Cart***
4. ***Place Orders & Make Payments***
5. ***Track Order Status***
6. ***Write Reviews for Ordered Items***

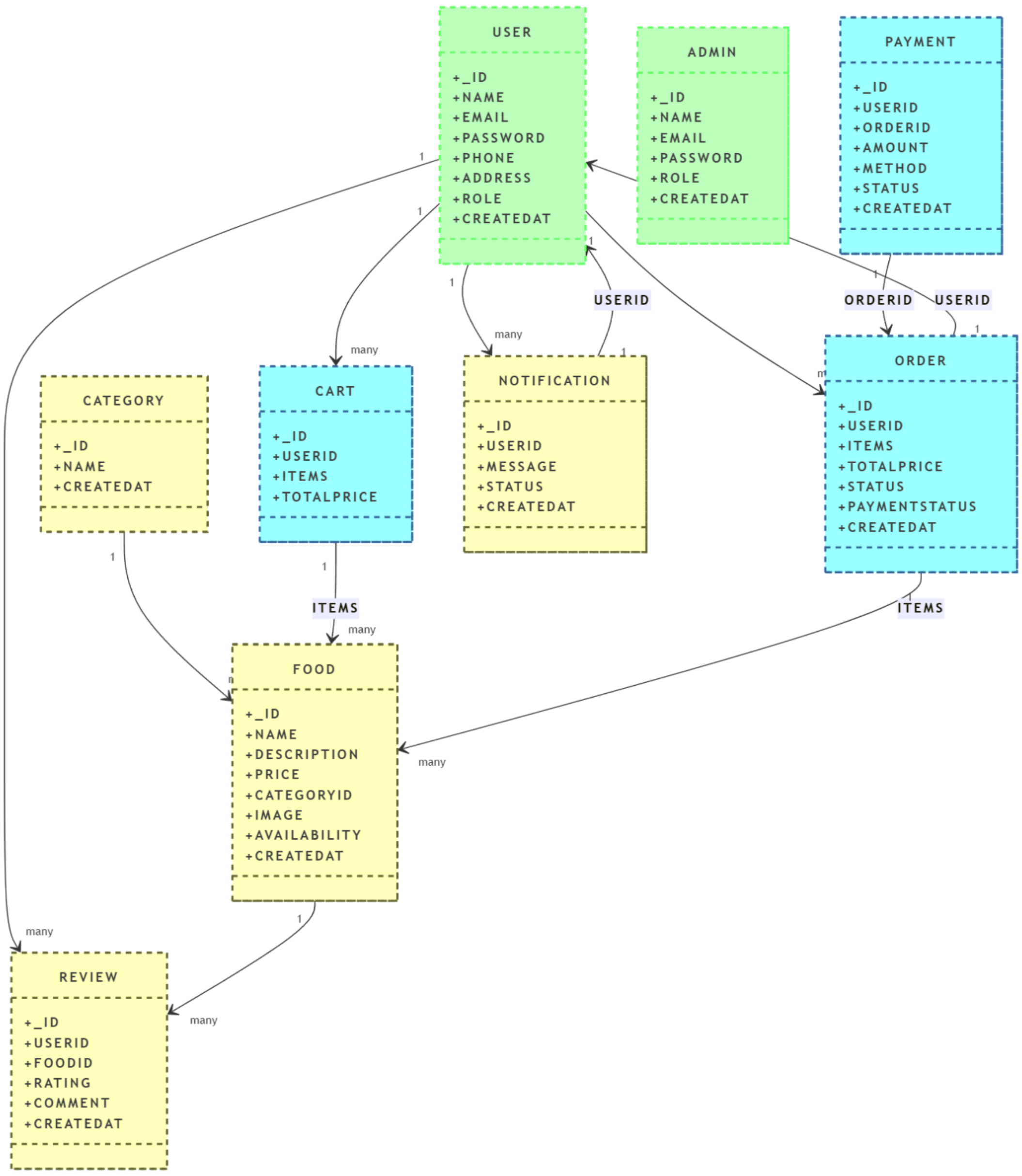
Step 5: Payment Gateway Integration

1. *Use **Razorpay** for online payments*
2. *Implement Webhook for payment verification*
3. *Store payment details in `payments` collection*

Step 6: Deploy the Application

1. *Backend → Deploy on **Vercel or Render***
2. *Database → Deploy **MongoDB Atlas***

--------------- Schema Diagram -------------


USER
+_ID
+NAME
+EMAIL
+PASSWORD
+PHONE
+ADDRESS
+ROLE
+CREATEDAT
ADMIN
+_ID
+NAME
+EMAIL
+PASSWORD
+ROLE
+CREATEDAT
PAYMENT
+_ID
+USERID
+ORDERID
+AMOUNT
+METHOD
+STATUS
+CREATEDAT
USERID
ORDERID
USERID
CATEGORY
+_ID
+NAME
+CREATEDAT
CART
+_ID
+USERID
+ITEMS
+TOTALPRICE
NOTIFICATION
+_ID
+USERID
+MESSAGE
+STATUS
+CREATEDAT
ORDER
+_ID
+USERID
+ITEMS
+TOTALPRICE
+STATUS
+PAYMENTSTATUS
+CREATEDAT
ITEMS
ITEMS
FOOD
+_ID
+NAME
+DESCRIPTION
+PRICE
+CATEGORYID
+IMAGE
+AVAILABILITY
+CREATEDAT
REVIEW
+_ID
+USERID
+FOODID
+RATING
+COMMENT
+CREATEDAT
1
many