**Project Title: Online Food Order Application**

**Name: K. Tejusree**

**Batch\_code: ANP-C7971**

**ID: AF0400950**

**Online Food Order Application**

**Introduction:**

This application focuses on creating an easy-to-use online food ordering app. In a world where convenience matters, our app will allow users to browse local restaurants, order their favorite meals, and pay securely—all from their smart phones. With features like real-time tracking and a simple interface, we aim to make ordering food quick and enjoyable, connecting people with delicious options at their fingertips.

**Abstract:**

This project presents an online food ordering application designed to simplify the process of ordering food from local restaurants through online. Users can easily browse menus, select their favorite dishes, and place orders using a user-friendly interface. The app includes features like secure payment options and real-time order tracking, ensuring a smooth experience from start to finish. By connecting customers with nearby restaurants, our application aims to make dining more convenient and accessible, catering to the growing demand for food delivery services.

**Entity Relationship Model:**

The Entity Relationship Model (ERM) for the online food order application consists of the following key entities, their attributes and the relationships between them, this model provides a clear and structured framework for designing and implementing the database.

**Entities:**

1. **User:** Represents customers who place orders.

* **Attributes:**

1. **UserID** (Primary Key)
2. Name
3. Email
4. Phone Number
5. Address
6. **Restaurant:** Represents the restaurants available for ordering.

* **Attributes:**
  1. **RestaurantID** (Primary Key)
  2. Name
  3. Location
  4. Cuisine Type
  5. Rating

1. **Menu Item:** Represents the individual food items offered by restaurants.

* **Attributes:**
  1. **ItemID** (Primary Key)
  2. Name
  3. Description
  4. Price
  5. **RestaurantID** (Foreign Key)

1. **Order:** Represents the orders placed by users.

* **Attributes:**
  1. **OrderID** (Primary Key)
  2. **UserID** (Foreign Key)
  3. TotalPrice
  4. OrderStatus
  5. Timestamp

1. **Order Item:** Represents the items in each order.

* **Attributes:**
  1. **OrderItemID** (Primary Key)
  2. **OrderID** (Foreign Key)
  3. **ItemID** (Foreign Key)
  4. Quantity

1. **Payment:** Represents payment details for orders.

* **Attributes:**
  1. **PaymentID** (Primary Key)
  2. **OrderID** (Foreign Key)
  3. Amount
  4. PaymentMethod
  5. PaymentStatus

### Reviews Table: Represents user-generated feedback about restaurants.

* **Attributes:**
* **ReviewID** (Primary Key)
* UserID (Foreign Key)
* RestaurantID (Foreign Key)
* Rating (1 to 5)
* Comment
* Timestamp

**Relationships:**

* **User** to **Order**: One-to-Many
  + A user can place multiple orders.
* **Restaurant** to **Menu Item**: One-to-Many
  + A restaurant can have multiple menu items.
* **Order** to **Order Item**: One-to-Many
  + An order can consist of multiple order items.
* **Order Item** to **Menu Item**: Many-to-One
  + Each order item corresponds to one menu item.
* **Order** to **Payment**: One-to-One
  + Each order has one payment associated with it.

This model helps organize the data flow within the application, ensuring smooth interactions between users, restaurants, and orders.

**Workflow:**

#### 1. User Registration and Login:

* **Sign Up**: Users create an account by entering their name, email, phone number, password, and address.
* **Log In**: Users enter their email and password to access their account.

#### 2. Browsing Restaurants:

* **Home Screen**: Users see a list of nearby restaurants.
* **Select Restaurant**: Users tap on a restaurant to view its menu and details.

#### 3. Exploring Menu Items:

* **View Menu**: Users browse available food items, seeing names, prices, and descriptions.
* **Add to Cart**: Users select items and specify quantities to add them to their cart.

#### 4. Cart Review:

* **Review Cart**: Users check their cart to see selected items and total price.
* **Modify Cart**: Users can change quantities or remove items if needed.
* **Checkout**: Users click “Checkout” to proceed.

#### 5. Order Placement:

* **Delivery Address**: Users confirm or enter their delivery address.
* **Payment Method**: Users choose how to pay (e.g., credit card, cash).
* **Order Summary**: Users review their order details before placing it.
* **Place Order**: Users click “Place Order” to confirm.

#### 6. Order Tracking:

* **Order Confirmation**: Users receive a notification that their order is confirmed.
* **Track Order**: Users can see the status of their order (e.g., being prepared, out for delivery).

#### 7. Order Delivery:

* **Delivery**: The delivery person picks up the order and heads to the user’s address.
* **Notify User**: Users receive a notification when their order is on the way.

#### 8. Payment Processing:

* **Payment**: If paying online, the transaction is processed at checkout. For cash on delivery, users pay the delivery person.

#### 9. Feedback and Ratings:

* **Rate Order**: After delivery, users can rate the restaurant and leave feedback.
* **Update Ratings**: Feedback is used to improve restaurant ratings in the app.

#### 10. User Profile Management:

* **Update Profile**: Users can change their details, like address and payment options.
* **Order History**: Users can view past orders and easily reorder favorites.

### Order from

**Restaurant**

**User**

**Order Items**

### Order

**Menu Items**

### Has

### Has

### Place Order

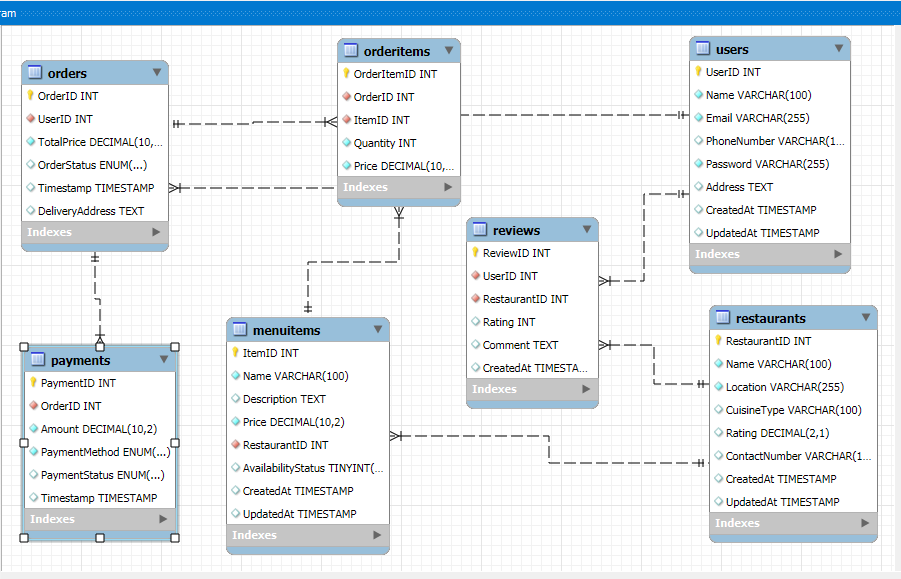
### Payments

**Orders**

**Reviews**

**ERR DIAGRAM**

**Database Design:**

****