Day 1 - *Restaurant Q-commerce Break Down of Tasks*

1. Marketplace Type:

- **Type**: General Q-Commerce
- **Purpose**: The goal of this marketplace is to be an e-commerce platform focused specifically in restaurant so that customers may browse the menu if the restaurant and easily order pizza, burgers, drinks, or deserts, It will simplify food ordering, ensuring delivery and take away, in a simple and user-friendly interface designed for only one restaurant.

2. Business Goals:

1. **What Problem Does Your Marketplace Aim To Solve?**

   The marketplace aims to solve the inconvenience of traditional food ordering by allowing customers to:

   - Eaily browse the restaurant's menu online.
   - Order without needing to call or visit the restaurant in person.
   - Customize their orders(e.g extra topping, spice levels)

2. **Who is Your Target Audience?**

   The target audinence is:

   - **Students:** seeking quick foot at a fair price.
   - **Office Workers :** looking for easy food around working hours
   - **Families:** Looking to have food delivered home at eat
   - **Local Community:** All people in the coverage area of the restaurant seeking quick and safe food service

3. **What Products/Services Will Be Offered?**
   - Food items: Pizza, burgers, fries, drinks and desserts.
   - Delivery and take away options.
   - Customization (e.g., extra cheese, extra salt, spicy etc.).
4. **How is Your Marketplace Unique?**

- **Speed**: Fast and reliable delivery service.
- **Affordability**: Competitive pricing on food items.
- **Customization**: Tailored food orders to meet individual preferences (e.g., spice levels, add-ons).

- **User-Friendly Interface**: A clean, intuitive platform for seamless navigation and ordering.
- **Real-Time Updates**: Live tracking of order preparation and delivery.

---

3. Data Schema Diagram:

1. **Products Data(Food Items)**:
    - `ProductID`: Unique identifier for each food item.
    - `Name`: Name of the food (e.g., Chicken Pizza, Chicken Burgur).
    - `slug`: Name of the food item to identify in dynamic routing (e.g., chicken-pizza, chicken-burgur).
    - `Price`: Price of the food item (e.g., $700).
    - `Category`: Type of food (e.g., Pizza, Burger, Drinks).
    - `Stock`: Whether the food is available or sold out.
2. **Order Data**:
    - `OrderID`: Unique identifier for each order.
    - `CustomerID`: Reference to the customer who placed the order.
    - `OrderDate`: Date and time when the order was placed.
    - `TotalAmount`: Total price of the order.
    - `OrderStatus`: Current status (e.g., Pending, Delivered).
3. **Customer Data**:
    - `CustomerID`: Unique identifier for each customer.
    - `Name`: Customer's name.
    - `Email`: Customer's email address.
    - `PhoneNumber`: Contact number.
    - `Address`: Delivery address.
4. **Delivery Zone**:
    - `ZoneID`: Unique identifier for the delivery zone..
    - `zoneName`: Name of the delivery zone.
    - `coverageArea`: area covered

Extra Thing:

- Each **Customer** can place multiple **Orders**.
- Each **Order** can include multiple **Products**..

---

5. Data Schema Example:

- **Product Data{**
    - ProductId:131123,
    - ProductName:"Chicken Pizza",
    - Slug:"chicken-pizza,
    - Price:250
    - Stock:200

}

- **Orders{**
    - OrderId:9123131, (Unique Id of every order)
    - CustomerId:91314141, (Unique id of every customer)
    - OrderDate:15/5/2025, (Order date)
    - TotalAmount : 500, (Total Amount Of cart items)
    - orderStaus: “pending || recived” ,**}**
- **Customer Data{**
    - CutomerId:1312 (Unique id of every customer)
    - CustomerName: ”Ahmed”,
    - CustomerEmail: abc@gmail.com,
    - phoneNumber : 03xxxxxx,
    - Address : ”etc ”

}

- **Delivery Zone{**
    - `ZoneID`: Unique identifier for the delivery zone..
    - `zoneName`:[Korangi , saddar , north karachi, malir].
    - `coverageArea`: All over the Karachi

}

3. Relationships Between Entities

1. **Products** → Connected to **Orders** through `ProductID`.
    - A single product can appear in multiple orders.
2. **Orders** → Connected to **Customers** through `CustomerID`.
    - A customer can place multiple orders.
3. **Orders** → Optionally connected to **Delivery Zones** through `ZoneID`.
    - Each order will fall within a delivery zone.

Schema Diagram

[Products]

- ProductID

- Name

- Slug

- Price

- Category

- Stock

|

|

[Orders] ---------> [Customers]

- OrderID - CustomerID

- CustomerID - Name

- OrderDate - Email

- TotalAmount - PhoneNumber

- OrderStatus - Address

|

|

[Delivery Zones]

- ZoneID

- ZoneName

- CoverageArea

- AssignedDrivers