Online Food Ordering System:

- Create a program that simulates an online food ordering system. Users can browse restaurants, view menus, and place orders for delivery or pickup

introduction:

- An online food ordering system allows your business to accept and manage orders placed online for delivery or takeaway.
- Customers browse a digital menu, either on an app or website and place and pay for their order online.
- Our online ordering system will help you transform your website into a money-making machine.
- No matter how much your business grows, you will always be able to take free unlimited orders with zero costs.
- Power your business with our free restaurant online ordering system & you’ll never have to worry about fees or commissions.

In [1]:

```
class Restaurant:
    def __init__(self, name, cuisine):
        self.name = name
        self.cuisine = cuisine
        self.menu = []

    def add_item_to_menu(self, item):
        self.menu.append(item)


class MenuItem:
    menu = {
        "pizza": 3.00,
        "nachos": 4.50,
        "popcorn": 6.00,
        "fries": 2.50,
        "chips": 1.00,
        "pretzel": 5.00,
        "soda": 3.00,
        "lemonade": 4.25
    }

    def display_menu(self):
        if not self.menu:
            print("Menu is empty.")
            return
        print("---------MENU----------")
        for key, value in self.menu.items():
            print(f"{key:10}:${value:.2f}")
        print("-----------------------")

    def get_user_order(self):
        cart = []
        while True:
            food = input("Select an item (q to quit): ").lower()
            if food == "q":
                break
            elif self.menu.get(food) is not None:
                cart.append(food)
        return cart
```

```
class Order:
    def __init__(self, restaurant, items, delivery, address):
        self.restaurant = restaurant
        self.items = items
        self.delivery = delivery
        self.address = address

    def calculate_total(self):
        total = 0
        for food in self.items:
            total += MenuItem.menu.get(food)
        return total


class OnlineFoodOrderingSystem:
    def __init__(self):
        self.restaurants = []

    def add_restaurant(self, restaurant):
        self.restaurants.append(restaurant)

    def browse_restaurants(self):
        print("Available Restaurants:")
        for i, restaurant in enumerate(self.restaurants):
            print(f"{i+1}. {restaurant.name}")

        selection = int(input("Select a restaurant: "))
        if 1 <= selection <= len(self.restaurants):
            return self.restaurants[selection - 1]
        else:
            print("Invalid selection.")
            return None

    def place_order(self, restaurant, items, delivery):
        if delivery:
            address = input("Enter delivery address: ")
        else:
            address = ""

        order = Order(restaurant, items, delivery, address)
        total = order.calculate_total()
```

```
        print("Order placed successfully!")
        print(f"Restaurant: {restaurant.name}")
        print("Items:")
        for item in items:
            print(f"- {item}: ${MenuItem.menu.get(item):.2f}")
        print(f"Delivery: {'Yes' if delivery else 'No'}")
        if delivery:
            print(f"Address: {address}")
        print(f"Total: ${total:.2f}")


# Creating restaurants and adding them to the system
system = OnlineFoodOrderingSystem()

restaurant1 = Restaurant("Restaurant A", "Italian")
restaurant1.add_item_to_menu(MenuItem())
restaurant1.add_item_to_menu(MenuItem())
system.add_restaurant(restaurant1)

restaurant2 = Restaurant("Restaurant B", "Chinese")
restaurant2.add_item_to_menu(MenuItem())
restaurant2.add_item_to_menu(MenuItem())
system.add_restaurant(restaurant2)

# Simulating user interaction
selected_restaurant = system.browse_restaurants()
if selected_restaurant is not None:
    menu_item = MenuItem()
    menu_item.display_menu()  # Display menu only once
    print()
    selected_items = menu_item.get_user_order()
    delivery = True if input("Do you want delivery? (yes/no): ").lower() == "yes" else False
    system.place_order(selected_restaurant, selected_items, delivery)
```

```
Available Restaurants:
1. Restaurant A
2. Restaurant B
Select a restaurant: 2
---------MENU----------
pizza     :$3.00
nachos    :$4.50
popcorn   :$6.00
fries     :$2.50
chips     :$1.00
pretzel   :$5.00
soda      :$3.00
lemonade  :$4.25
------------------------

Select an item (q to quit): popcorn
Select an item (q to quit): pizza
Select an item (q to quit): chips
Select an item (q to quit): fries
Select an item (q to quit): q
Do you want delivery? (yes/no): yes
Enter delivery address: door_no:4-23,bcom colony,dharmavaram,ananthapur,515671
Order placed successfully!
Restaurant: Restaurant B
Items:
- popcorn: $6.00
- pizza: $3.00
- chips: $1.00
- fries: $2.50
Delivery: Yes
Address: door_no:4-23,bcom colony,dharmavaram,ananthapur,515671
Total: $12.50
```