```
# Inventory Management System
inventory = {
    "Apple": {"category": "Fruits", "unit": "kg", "stock": 50, "price": 150},
    "Orange": {"category": "Fruits", "unit": "kg", "stock": 40, "price": 120},
    "Potato": {"category": "Vegetables", "unit": "kg", "stock": 100, "price": 25},
    "Rice": {"category": "Grains", "unit": "kg", "stock": 200, "price": 50},
    "Milk": {"category": "Dairy", "unit": "liter", "stock": 100, "price": 60},
    "Biscuits": {"category": "Packaged Foods", "unit": "pack", "stock": 60, "price": 30},
}

def display_inventory():
    print("\nCurrent Inventory:")
    print(f"{'Item':<12}{'Category':<20}{'Stock':<10}{'Unit':<10}{'Price (₹/Unit)':<15}")
    print("-" * 60)
    for item, details in inventory.items():

print(f"{item:<12}{details['category']:<20}{details['stock']:<10}{details['unit']:<10}{details['price']:<1
5}")
    print("-" * 60)

def place_order():
    print("\nPlace your order. Type 'done' when finished.")
    order = {}
    while True:
        item_name = input("Enter item name (or 'done' to finish): ").strip()
        if item_name.lower() == "done":
            break

        if item_name not in inventory:
            print("Item not found in inventory. Please enter a valid item.")
            continue

        try:
            quantity = float(input(f"Enter quantity of {item_name} ({inventory[item_name]['unit']}): "))
            if quantity <= 0:
                print("Quantity must be a positive number. Try again.")
                continue
        except ValueError:
            print("Invalid quantity. Please enter a number.")
            continue

        available_stock = inventory[item_name]["stock"]
        if quantity > available_stock:
            print(f"Only {available_stock} {inventory[item_name]['unit']} of {item_name} is available.")
            quantity = available_stock

        order[item_name] = order.get(item_name, 0) + quantity

    return order

def process_order(order):
    print("\nOrder Summary:")
    total_cost = 0
    total_weight = 0

    for item_name, quantity in order.items():
        unit_price = inventory[item_name]["price"]
        category = inventory[item_name]["category"]
        unit = inventory[item_name]["unit"]

        cost = quantity * unit_price
        total_cost += cost
        if unit == "kg":
            total_weight += quantity

        # Update inventory
        inventory[item_name]["stock"] -= quantity

        print(f"{item_name}: {quantity} {unit} × ₹{unit_price}/{unit} = ₹{cost}")

    print(f"\nTotal Weight: {total_weight} kg")
    print(f"Total Cost: ₹{total_cost}")

    # Display updated inventory
    display_inventory()

    # Highlight low-stock warnings
    print("\nLow-Stock Warnings:")
    for item_name, details in inventory.items():
        if details["stock"] < 5:
            print(f"⚠️  {item_name} is low on stock ({details['stock']} {details['unit']} remaining).")

def main():
    print("Welcome to the Grocery Shop Inventory Management System!")
    display_inventory()
    order = place_order()
    if order:
        process_order(order)
    else:
        print("\nNo items ordered. Exiting.")

if __name__ == "__main__":
    main()
```