





```

In [13]: import datetime
import pytz
import json
import os
import xlswriter
from reportlab.pdfgen import canvas

class Order:
    order_counter = 0

    def __init__(self, customer_name, items, quantities):
        Order.order_counter += 1
        self.order_id = Order.order_counter
        self.customer_name = customer_name
        self.items = items
        self.quantities = quantities
        self.order_date = datetime.datetime.now(pytz.timezone('Asia/Kolkata')).strftime("%Y-%m-%d %H:%M:%S")

    def display_order_details(self, menu):
        print("\nOrder Details:")
        print("Order ID:", self.order_id)
        print("Customer Name:", self.customer_name)
        print("Order Date:", self.order_date)
        print("\nItems:")
        total_price = 0

        for item, quantity in zip(self.items, self.quantities):
            price = menu[item]
            item_total = price * quantity
            total_price += item_total
            print(f"{item} - Quantity: {quantity} - Price: {price} - Total: {item_total}")

        print("\nTotal Price:", total_price)

    def to_dict(self):
        return {
            'order_id': self.order_id,
            'customer_name': self.customer_name,
            'items': self.items,
            'quantities': self.quantities,
            'order_date': self.order_date
        }

class BakeryManagementSystem:
    def __init__(self):
        self.orders = []
        self.menu = {
            'Pizza': 8.50,
            'Burger': 5.00,
            'Cake': 20.00,
            'Cookies': 2.50,
            'Croissant': 3.00,
            'Muffin': 4.00
        }
        # Add more items as needed with their respective prices

    def load_order_history(self):
        if os.path.exists('order_history.json'):
            with open('order_history.json', 'r') as file:
                order_data = json.load(file)
                self.orders = [Order(customer_name=order['customer_name'],
                                     items=order['items'],
                                     quantities=order['quantities']) for order in order_data]

    def save_order_history(self):
        order_data = [order.to_dict() for order in self.orders]
        with open('order_history.json', 'w') as file:
            json.dump(order_data, file, indent=4)

```

```

# ...

# ...

def export_to_excel(self):
    if not self.orders:
        print("No orders to export.")
        return

    workbook = xlswriter.Workbook('order_history.xlsx')
    worksheet = workbook.add_worksheet()

    headers = ['Order ID', 'Customer Name', 'Items', 'Quantities', 'Order Date']
    for col, header in enumerate(headers):
        worksheet.write(0, col, header)

    for row, order in enumerate(self.orders, start=1):
        worksheet.write(row, 0, order.order_id)
        worksheet.write(row, 1, order.customer_name)
        worksheet.write(row, 2, ', '.join(order.items))
        worksheet.write(row, 3, ', '.join(map(str, order.quantities)))
        worksheet.write(row, 4, order.order_date)

    workbook.close()
    print("Order history exported to Excel successfully!")

def export_to_pdf(self, order_id):
    for order in self.orders:
        if order.order_id == order_id:
            pdf_filename = f'bill_order_{order_id}.pdf'
            canvas_obj = canvas.Canvas(pdf_filename)

            canvas_obj.drawString(72, 800, "Order Bill")
            canvas_obj.drawString(72, 780, f"Order ID: {order.order_id}")
            canvas_obj.drawString(72, 760, f"Customer Name: {order.customer_name}")
            canvas_obj.drawString(72, 740, f"Order Date: {order.order_date}")

            y_position = 720
            for item, quantity in zip(order.items, order.quantities):
                price = self.menu[item]
                item_total = price * quantity
                canvas_obj.drawString(72, y_position, f"{item} - Quantity: {quantity} -")
                y_position -= 20

            total_price = sum(self.menu[item] * quantity for item, quantity in zip(order.items, order.quantities))
            canvas_obj.drawString(72, y_position, f"Total Price: {total_price}")

            canvas_obj.save()
            print(f"Bill exported to {pdf_filename}")
            return
    print("Order not found.")

def add_order(self, customer_name):
    items = []
    quantities = []

    print("\nMenu:")
    for item, price in self.menu.items():
        print(f"{item} - ${price}")

    while True:
        item = input("Enter item from the menu (or 'done' to finish): ")
        if item.lower() == 'done':
            break

        if item in self.menu:
            quantity = int(input(f"Enter quantity for {item}: "))
            items.append(item)

```

```

        quantities.append(quantity)
    else:
        print("Invalid item. Please choose from the menu.")

    new_order = Order(customer_name, items, quantities)
    self.orders.append(new_order)
    print("\nOrder added successfully!")
    return new_order.order_id

def get_order_details(self, order_id):
    for order in self.orders:
        if order.order_id == order_id:
            order.display_order_details(self.menu)
            return
    print("Order not found.")

def modify_order(self, order_id):
    for order in self.orders:
        if order.order_id == order_id:
            print("\nCurrent Order Details:")
            order.display_order_details(self.menu)

            items = []
            quantities = []

            print("\nMenu:")
            for item, price in self.menu.items():
                print(f"{item} - ${price}")

            while True:
                item = input("Enter item from the menu to modify (or 'done' to finish): ")
                if item.lower() == 'done':
                    break

                if item in self.menu:
                    quantity = int(input(f"Enter new quantity for {item}: "))
                    items.append(item)
                    quantities.append(quantity)
                else:
                    print("Invalid item. Please choose from the menu.")

            order.items = items
            order.quantities = quantities
            order.order_date = datetime.datetime.now(pytz.timezone('Asia/Kolkata')).strftime("%Y-%m-%d %H:%M:%S")
            print("\nOrder modified successfully!")
            return
    print("Order not found.")

def main():
    bakery_system = BakeryManagementSystem()
    bakery_system.load_order_history()

    while True:
        print("\nBakery Management System")
        print("1. Add Order")
        print("2. Get Order Details")
        print("3. Modify Order")
        print("4. Export Order History to Excel")
        print("5. Export Bill to PDF")
        print("6. Exit")

        choice = input("Enter your choice (1-6): ")

        if choice == "1":
            customer_name = input("Enter Customer Name: ")
            order_id = bakery_system.add_order(customer_name)
            print(f"Order ID: {order_id}")

        elif choice == "2":
            order_id = int(input("Enter Order ID to retrieve details: "))
            bakery_system.get_order_details(order_id)

```

```
elif choice == "3":
    order_id = int(input("Enter Order ID to modify: "))
    bakery_system.modify_order(order_id)

elif choice == "4":
    bakery_system.export_to_excel()
    print("Order history exported to Excel successfully!")

elif choice == "5":
    order_id = int(input("Enter Order ID to export bill to PDF: "))
    bakery_system.export_to_pdf(order_id)

elif choice == "6":
    bakery_system.save_order_history()
    print("Exiting Bakery Management System. Order history saved. Goodbye!")
    break

else:
    print("Invalid choice. Please enter a number between 1 and 6.")

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

Bakery Management System

1. Add Order
2. Get Order Details
3. Modify Order
4. Export Order History to Excel
5. Export Bill to PDF
6. Exit

Enter your choice (1-6): 1

Enter Customer Name: Bhavesh

Menu:

Pizza - \$8.5

Burger - \$5.0

Cake - \$20.0

Cookies - \$2.5

Croissant - \$3.0

Muffin - \$4.0

Enter item from the menu (or 'done' to finish): Pizza

Enter quantity for Pizza: 10

Enter item from the menu (or 'done' to finish): Burger

Enter quantity for Burger: 10

Enter item from the menu (or 'done' to finish): Cookies

Enter quantity for Cookies: 10

Enter item from the menu (or 'done' to finish): done

Order added successfully!

Order ID: 2

Bakery Management System

1. Add Order
2. Get Order Details
3. Modify Order
4. Export Order History to Excel
5. Export Bill to PDF
6. Exit

Enter your choice (1-6): 2

Enter Order ID to retrieve details: 2

Order Details:

Order ID: 2

Customer Name: Bhavesh

Order Date: 2024-01-28 13:29:40

Items:

Pizza - Quantity: 10 - Price: 8.5 - Total: 85.0

Burger - Quantity: 10 - Price: 5.0 - Total: 50.0

Cookies - Quantity: 10 - Price: 2.5 - Total: 25.0

Total Price: 160.0

Bakery Management System

1. Add Order
2. Get Order Details
3. Modify Order
4. Export Order History to Excel
5. Export Bill to PDF
6. Exit

Enter your choice (1-6): 5

Enter Order ID to export bill to PDF: 2

Bill exported to bill\_order\_2.pdf

Bakery Management System

1. Add Order
2. Get Order Details
3. Modify Order
4. Export Order History to Excel
5. Export Bill to PDF
6. Exit

Enter your choice (1-6): 4

Order history exported to Excel successfully!  
Order history exported to Excel successfully!

Bakery Management System

1. Add Order
2. Get Order Details
3. Modify Order
4. Export Order History to Excel
5. Export Bill to PDF
6. Exit

Enter your choice (1-6): 6

Exiting Bakery Management System. Order history saved. Goodbye!