INTERACTIVE FOOD ORDERING SYSTEM

Suhani Yalaga

INTRODUCTION

- Objective is to develop a straightforward food ordering system using Python. This system will enable users to effortlessly browse menus, choose their desired items, complete their orders, and make payments, upon completing the order, essential details such as the order list, total amount, payment method, customer's name, and address will be automatically stored. aim is to simplify the food ordering process for all users.

USER FRIENDLY

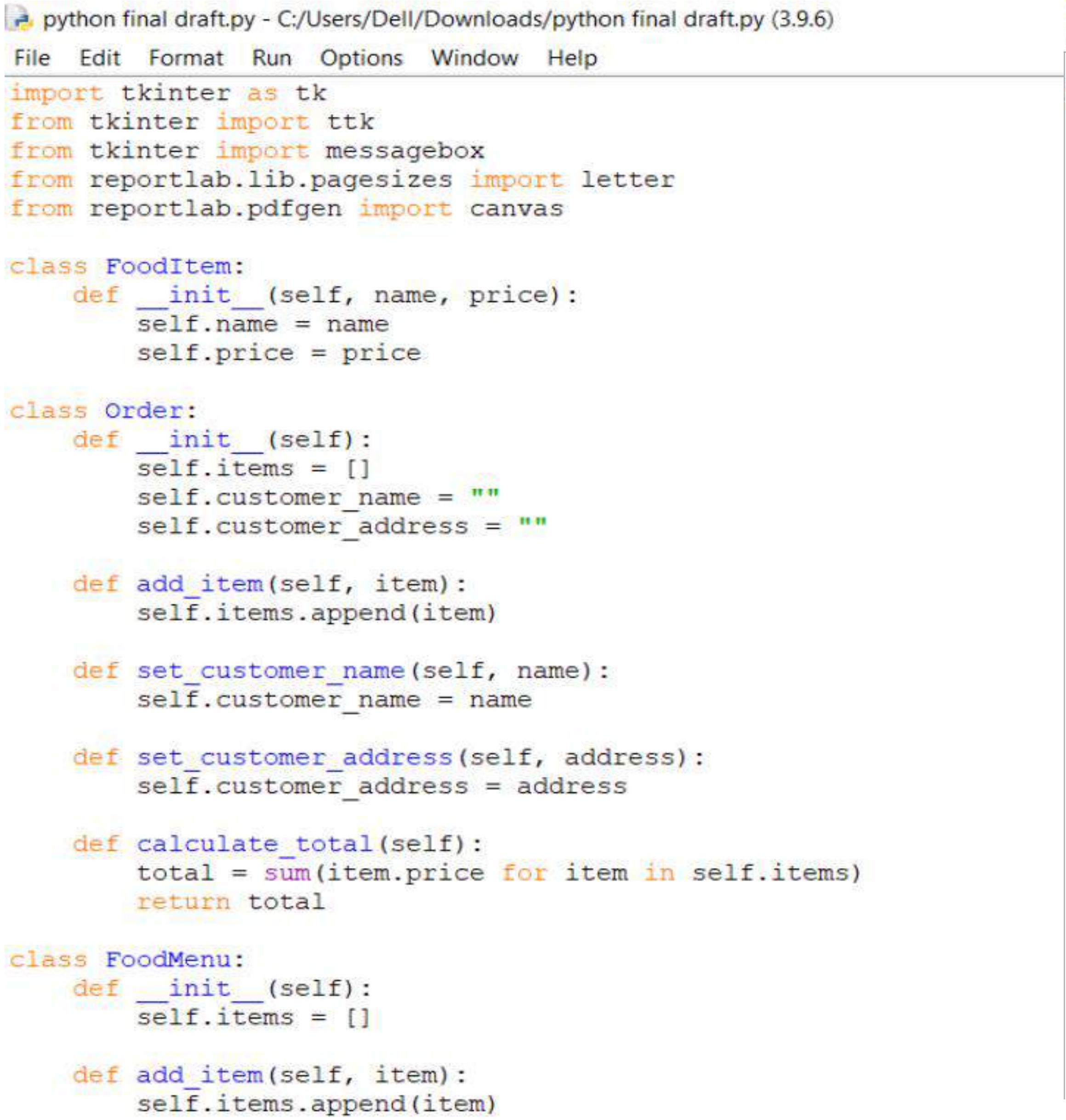
BENEFITS

- **Customers :**
- Effortlessly navigate through menus: Users can easily browse through the available food items, view descriptions, and make selections.
- Select items and place orders: Customers can add items to their virtual cart and place orders with just a few clicks or taps, eliminating the need for manual order-taking.
- Make payments: The system facilitates secure payment transactions, enabling customers to complete their orders seamlessly.
- **Restaurants staff:**
- Generation of PDF reports: The system automatically generates PDF reports for placed orders, aiding in order tracking, inventory management, and financial reporting.

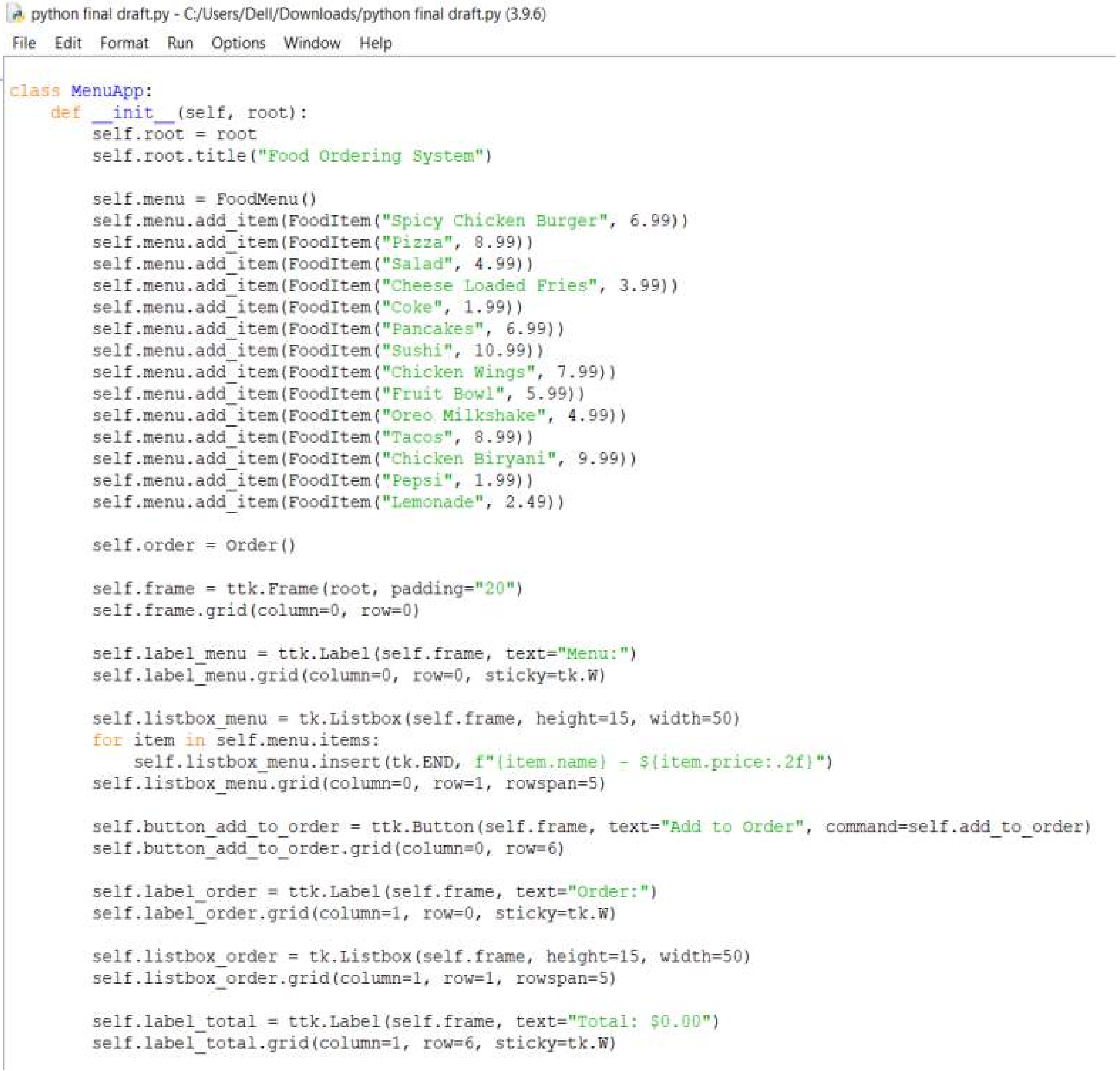
PYTHON LIBRARIES USED:

- tkinter: This library is used for creating the graphical user interface (GUI).
- tk.messagebox: Used for displaying message boxes in the GUI.
- reportlab.pdfgen. These are used for generating PDF files.
- reportlab.lib.pagesizes Specifies the page size for the PDF document as a standard letter size.

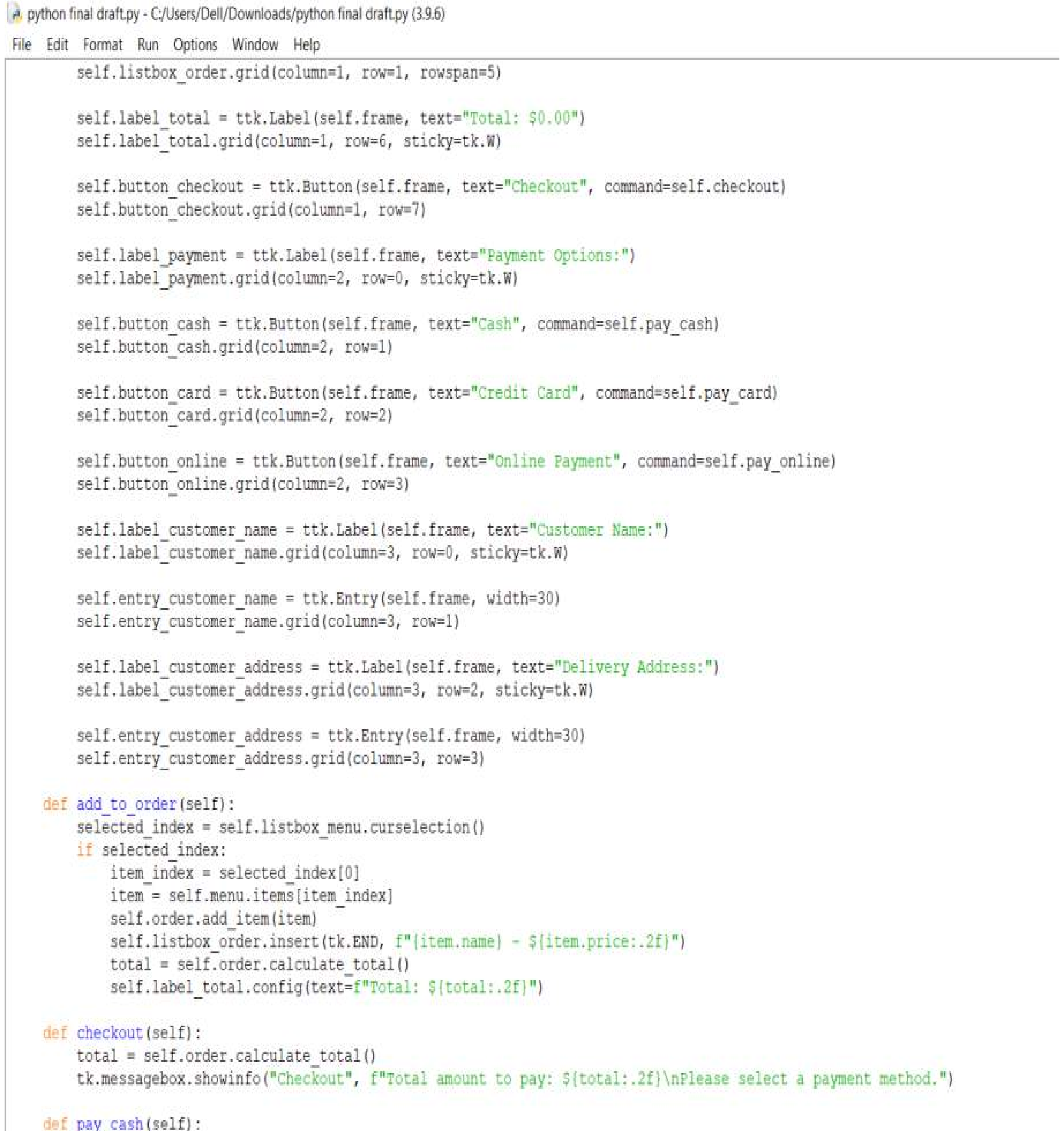
- **Code**


python final draft.py - C:/Users/Dell/Downloads/python final draft.py (3.9.6)
File Edit Format Run Options Window Help

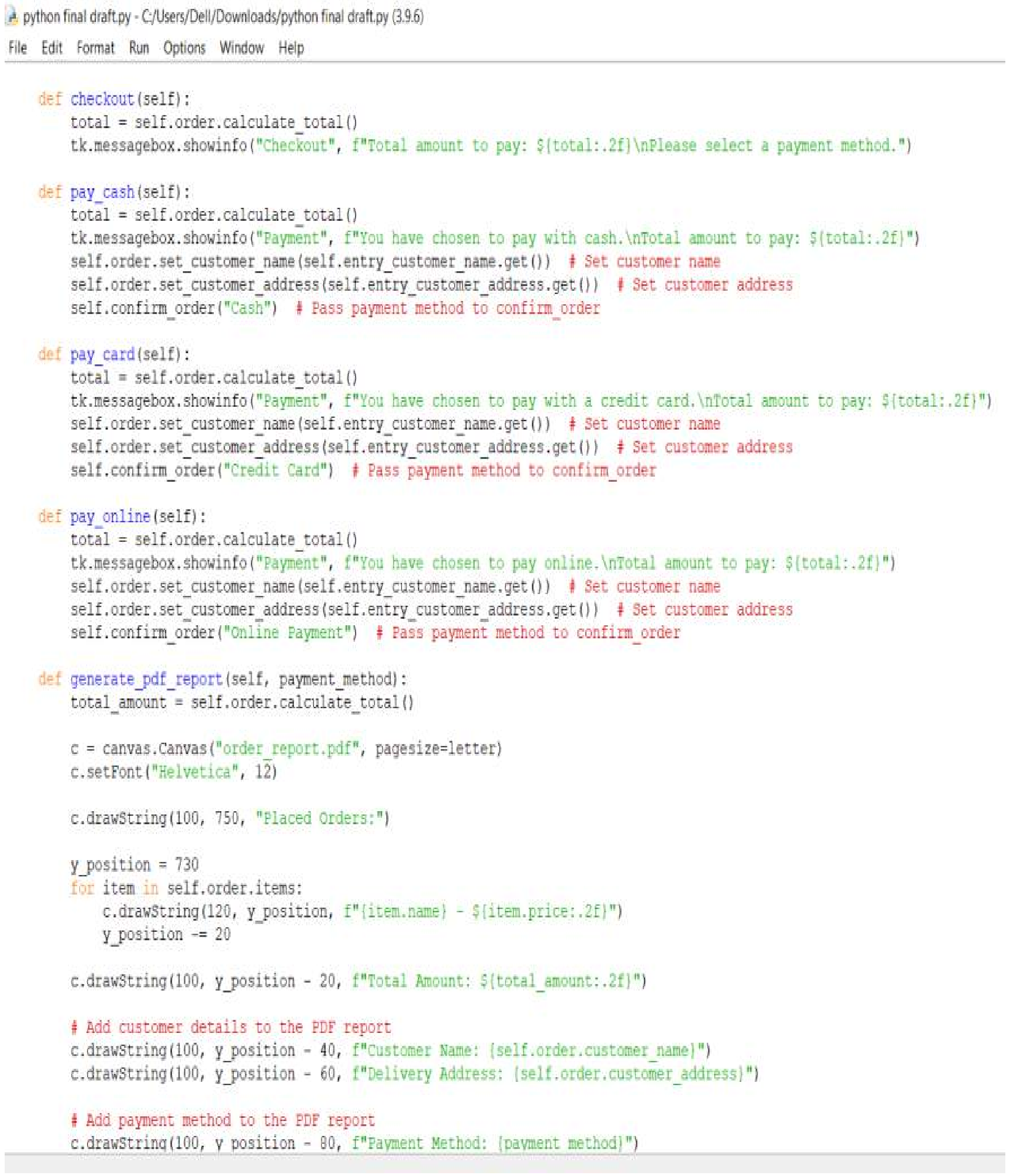
```python
import tkinter as tk
from tkinter import ttk
from tkinter import messagebox
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas

class FoodItem:
    def __init__(self, name, price):
        self.name = name
        self.price = price

class Order:
    def __init__(self):
        self.items = []
        self.customer_name = ""
        self.customer_address = ""

    def add_item(self, item):
        self.items.append(item)

    def set_customer_name(self, name):
        self.customer_name = name

    def set_customer_address(self, address):
        self.customer_address = address

    def calculate_total(self):
        total = sum(item.price for item in self.items)
        return total

class FoodMenu:
    def __init__(self):
        self.items = []

    def add_item(self, item):
        self.items.append(item)
```


python final draft.py - C:/Users/Dell/Downloads/python final draft.py (3.9.6)
File Edit Format Run Options Window Help

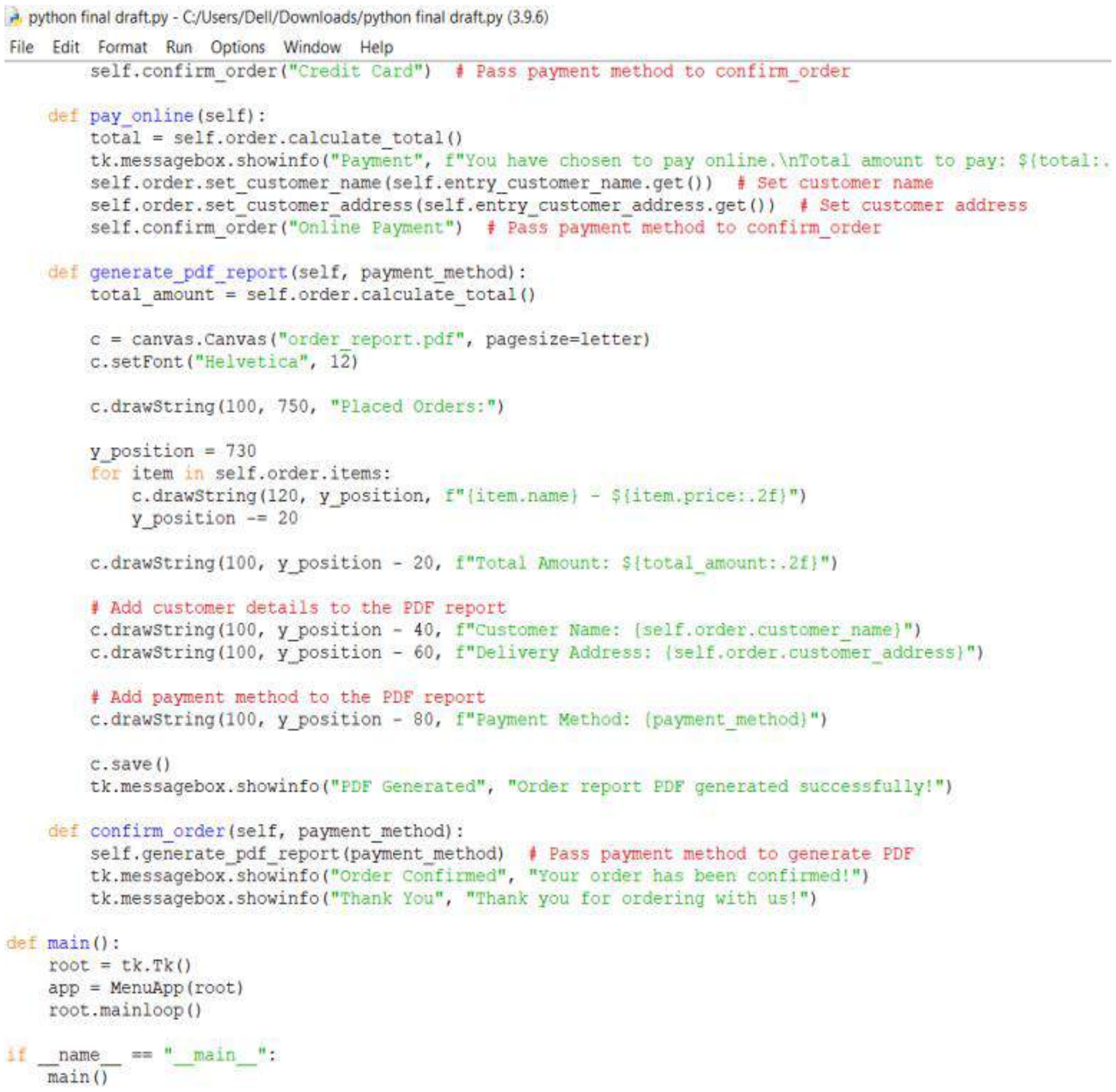
```python
class MenuApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Food Ordering System")

        self.menu = FoodMenu()
        self.menu.add_item(FoodItem("Spicy Chicken Burger", 6.99))
        self.menu.add_item(FoodItem("Pizza", 8.99))
        self.menu.add_item(FoodItem("Salad", 4.99))
        self.menu.add_item(FoodItem("Cheese Loaded Fries", 3.99))
        self.menu.add_item(FoodItem("Coke", 1.99))
        self.menu.add_item(FoodItem("Pancakes", 6.99))
        self.menu.add_item(FoodItem("Sushi", 10.99))
        self.menu.add_item(FoodItem("Chicken Wings", 7.99))
        self.menu.add_item(FoodItem("Fruit Bowl", 5.99))
        self.menu.add_item(FoodItem("Oreo Milkshake", 4.99))
        self.menu.add_item(FoodItem("Tacos", 8.99))
        self.menu.add_item(FoodItem("Chicken Biryani", 9.99))
        self.menu.add_item(FoodItem("Pepsi", 1.99))
        self.menu.add_item(FoodItem("Lemonade", 2.49))

        self.order = Order()

        self.frame = ttk.Frame(root, padding="20")
        self.frame.grid(column=0, row=0)

        self.label_menu = ttk.Label(self.frame, text="Menu:")
        self.label_menu.grid(column=0, row=0, sticky=tk.W)

        self.listbox_menu = tk.Listbox(self.frame, height=15, width=50)
        for item in self.menu.items:
            self.listbox_menu.insert(tk.END, f"{item.name} - ${item.price:.2f}")
        self.listbox_menu.grid(column=0, row=1, rowspan=5)

        self.button_add_to_order = ttk.Button(self.frame, text="Add to Order", command=self.add_to_order)
        self.button_add_to_order.grid(column=0, row=6)

        self.label_order = ttk.Label(self.frame, text="Order:")
        self.label_order.grid(column=1, row=0, sticky=tk.W)

        self.listbox_order = tk.Listbox(self.frame, height=15, width=50)
        self.listbox_order.grid(column=1, row=1, rowspan=5)

        self.label_total = ttk.Label(self.frame, text="Total: $0.00")
        self.label_total.grid(column=1, row=6, sticky=tk.W)
```


python final draft.py - C:/Users/Dell/Downloads/python final draft.py (3.9.6)
File Edit Format Run Options Window Help

```
        self.listbox_order.grid(column=1, row=1, rowspan=5)

        self.label_total = ttk.Label(self.frame, text="Total: $0.00")
        self.label_total.grid(column=1, row=6, sticky=tk.W)

        self.button_checkout = ttk.Button(self.frame, text="Checkout", command=self.checkout)
        self.button_checkout.grid(column=1, row=7)

        self.label_payment = ttk.Label(self.frame, text="Payment Options:")
        self.label_payment.grid(column=2, row=0, sticky=tk.W)

        self.button_cash = ttk.Button(self.frame, text="Cash", command=self.pay_cash)
        self.button_cash.grid(column=2, row=1)

        self.button_card = ttk.Button(self.frame, text="Credit Card", command=self.pay_card)
        self.button_card.grid(column=2, row=2)

        self.button_online = ttk.Button(self.frame, text="Online Payment", command=self.pay_online)
        self.button_online.grid(column=2, row=3)

        self.label_customer_name = ttk.Label(self.frame, text="Customer Name:")
        self.label_customer_name.grid(column=3, row=0, sticky=tk.W)

        self.entry_customer_name = ttk.Entry(self.frame, width=30)
        self.entry_customer_name.grid(column=3, row=1)

        self.label_customer_address = ttk.Label(self.frame, text="Delivery Address:")
        self.label_customer_address.grid(column=3, row=2, sticky=tk.W)

        self.entry_customer_address = ttk.Entry(self.frame, width=30)
        self.entry_customer_address.grid(column=3, row=3)

    def add_to_order(self):
        selected_index = self.listbox_menu.curselection()
        if selected_index:
            item_index = selected_index[0]
            item = self.menu.items[item_index]
            self.order.add_item(item)
            self.listbox_order.insert(tk.END, f"{item.name} - ${item.price:.2f}")
            total = self.order.calculate_total()
            self.label_total.config(text=f"Total: ${total:.2f}")

    def checkout(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Checkout", f"Total amount to pay: ${total:.2f}\nPlease select a payment method.")

    def pay_cash(self):
```

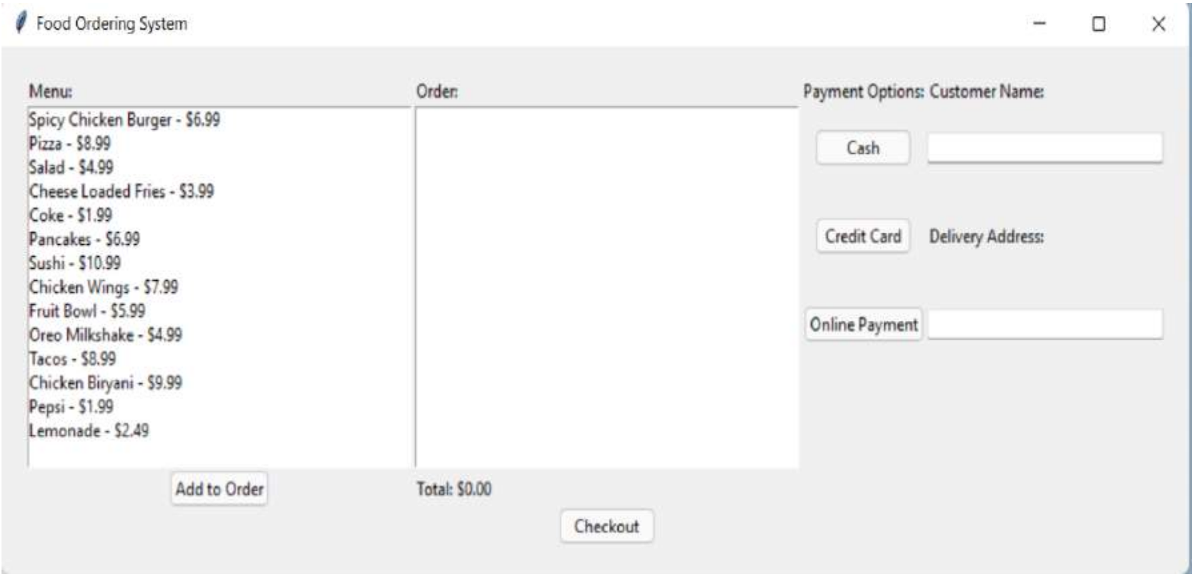

python final draft.py - C:/Users/Dell/Downloads/python final draft.py (3.9.6)
File Edit Format Run Options Window Help

```
    def checkout(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Checkout", f"Total amount to pay: ${total:.2f}\nPlease select a payment method.")

    def pay_cash(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Payment", f"You have chosen to pay with cash.\nTotal amount to pay: ${total:.2f}")
        self.order.set_customer_name(self.entry_customer_name.get())  # Set customer name
        self.order.set_customer_address(self.entry_customer_address.get())  # Set customer address
        self.confirm_order("Cash")  # Pass payment method to confirm_order

    def pay_card(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Payment", f"You have chosen to pay with a credit card.\nTotal amount to pay: ${total:.2f}")
        self.order.set_customer_name(self.entry_customer_name.get())  # Set customer name
        self.order.set_customer_address(self.entry_customer_address.get())  # Set customer address
        self.confirm_order("Credit Card")  # Pass payment method to confirm_order

    def pay_online(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Payment", f"You have chosen to pay online.\nTotal amount to pay: ${total:.2f}")
        self.order.set_customer_name(self.entry_customer_name.get())  # Set customer name
        self.order.set_customer_address(self.entry_customer_address.get())  # Set customer address
        self.confirm_order("Online Payment")  # Pass payment method to confirm_order

    def generate_pdf_report(self, payment_method):
        total_amount = self.order.calculate_total()

        c = canvas.Canvas("order_report.pdf", pagesize=letter)
        c.setFont("Helvetica", 12)

        c.drawString(100, 750, "Placed Orders:")

        y_position = 730
        for item in self.order.items:
            c.drawString(120, y_position, f"{item.name} - ${item.price:.2f}")
            y_position -= 20

        c.drawString(100, y_position - 20, f"Total Amount: ${total_amount:.2f}")

        # Add customer details to the PDF report
        c.drawString(100, y_position - 40, f"Customer Name: {self.order.customer_name}")
        c.drawString(100, y_position - 60, f"Delivery Address: {self.order.customer_address}")

        # Add payment method to the PDF report
        c.drawString(100, y_position - 80, f"Payment Method: {payment_method}")
```


python final draft.py - C:/Users/Dell/Downloads/python final draft.py (3.9.6)
File Edit Format Run Options Window Help

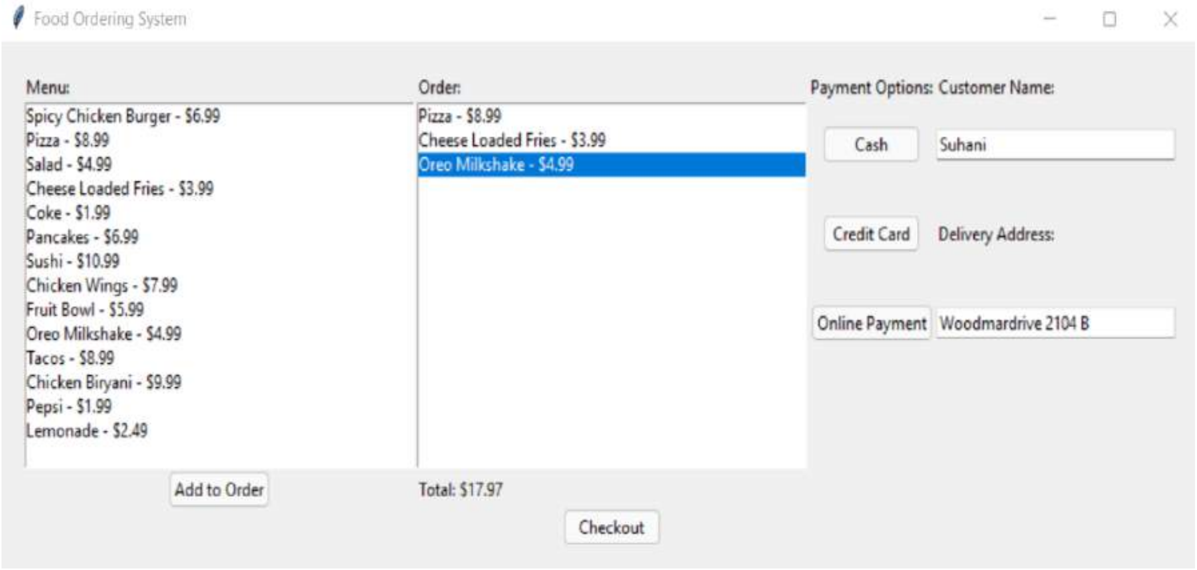
```
        self.confirm_order("Credit Card")  # Pass payment method to confirm_order

    def pay_online(self):
        total = self.order.calculate_total()
        tk.messagebox.showinfo("Payment", f"You have chosen to pay online.\nTotal amount to pay: ${total:.
        self.order.set_customer_name(self.entry_customer_name.get())  # Set customer name
        self.order.set_customer_address(self.entry_customer_address.get())  # Set customer address
        self.confirm_order("Online Payment")  # Pass payment method to confirm_order

    def generate_pdf_report(self, payment_method):
        total_amount = self.order.calculate_total()

        c = canvas.Canvas("order_report.pdf", pagesize=letter)
        c.setFont("Helvetica", 12)

        c.drawString(100, 750, "Placed Orders:")

        y_position = 730
        for item in self.order.items:
            c.drawString(120, y_position, f"{item.name} - ${item.price:.2f}")
            y_position -= 20

        c.drawString(100, y_position - 20, f"Total Amount: ${total_amount:.2f}")

        # Add customer details to the PDF report
        c.drawString(100, y_position - 40, f"Customer Name: {self.order.customer_name}")
        c.drawString(100, y_position - 60, f"Delivery Address: {self.order.customer_address}")

        # Add payment method to the PDF report
        c.drawString(100, y_position - 80, f"Payment Method: {payment_method}")

        c.save()
        tk.messagebox.showinfo("PDF Generated", "Order report PDF generated successfully!")

    def confirm_order(self, payment_method):
        self.generate_pdf_report(payment_method)  # Pass payment method to generate PDF
        tk.messagebox.showinfo("Order Confirmed", "Your order has been confirmed!")
        tk.messagebox.showinfo("Thank You", "Thank you for ordering with us!")

def main():
    root = tk.Tk()
    app = MenuApp(root)
    root.mainloop()

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

OUTPUT



Spicy Chicken Burger - $6.99
Pizza - $8.99
Salad - $4.99
Cheese Loaded Fries - $3.99
Coke - $1.99
Pancakes - $6.99
Sushi - $10.99
Chicken Wings - $7.99
Fruit Bowl - $5.99
Oreo Milkshake - $4.99
Tacos - $8.99
Chicken Biryani - $9.99
Pepsi - $1.99
Lemonade - $2.49

Total: $0.00



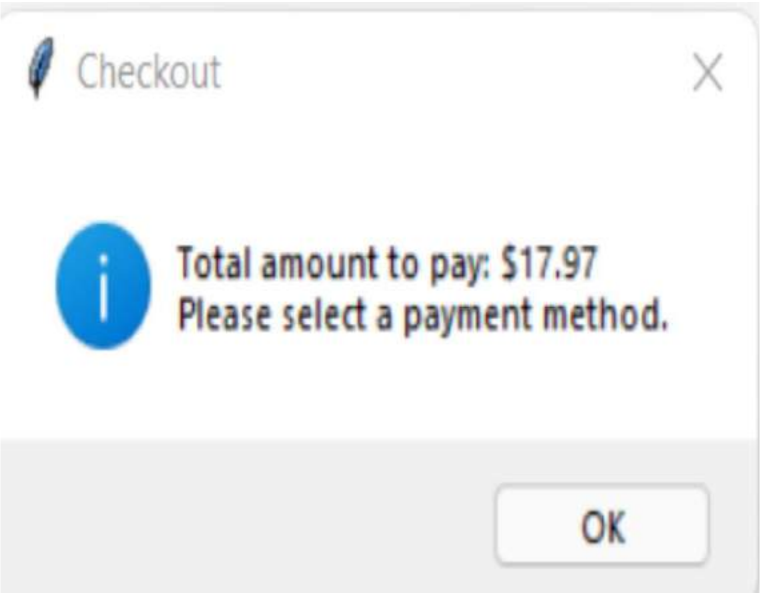
Spicy Chicken Burger - $6.99
Pizza - $8.99
Salad - $4.99
Cheese Loaded Fries - $3.99
Coke - $1.99
Pancakes - $6.99
Sushi - $10.99
Chicken Wings - $7.99
Fruit Bowl - $5.99
Oreo Milkshake - $4.99
Tacos - $8.99
Chicken Biryani - $9.99
Pepsi - $1.99
Lemonade - $2.49

Pizza - $8.99
Cheese Loaded Fries - $3.99
Oreo Milkshake - $4.99

Customer Name: Suhani

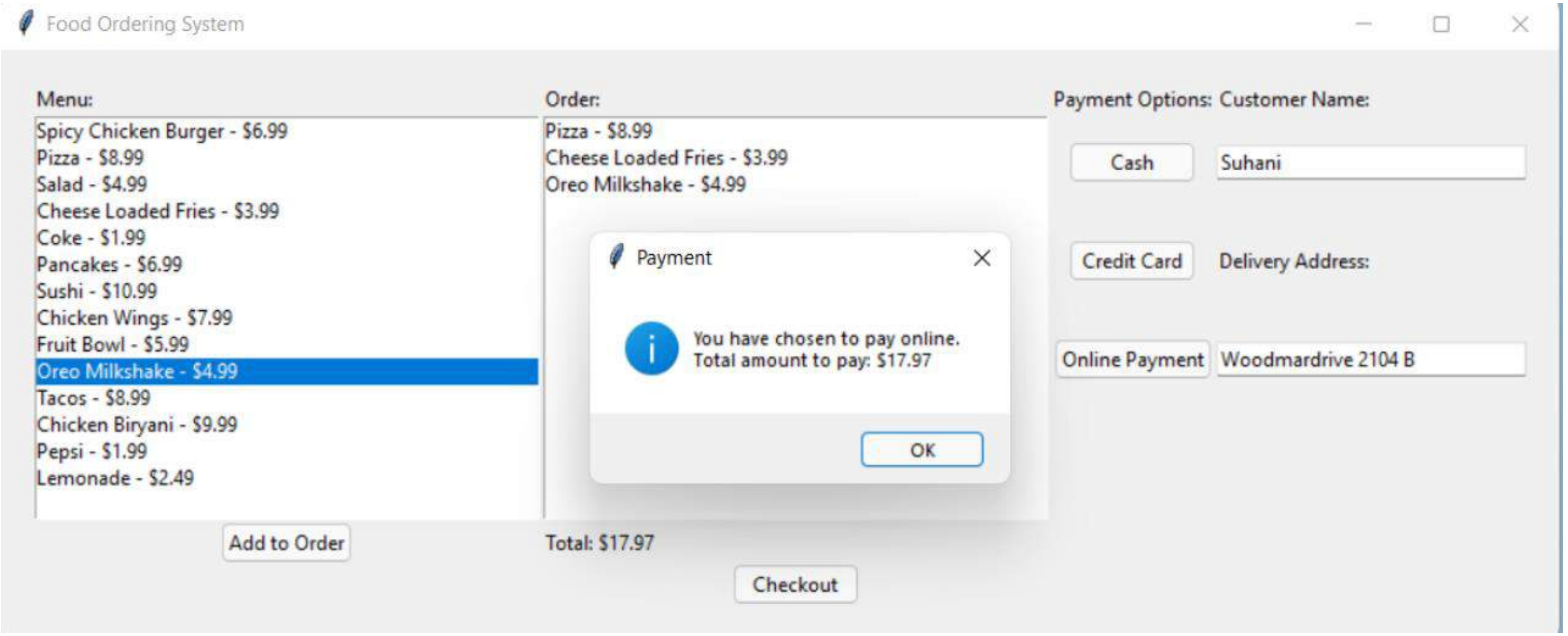
Woodmardrive 2104 B

Total: $17.97



Checkout

Total amount to pay: $17.97
Please select a payment method.



Food Ordering System

Menu:

Spicy Chicken Burger - $6.99
Pizza - $8.99
Salad - $4.99
Cheese Loaded Fries - $3.99
Coke - $1.99
Pancakes - $6.99
Sushi - $10.99
Chicken Wings - $7.99
Fruit Bowl - $5.99
Oreo Milkshake - $4.99
Tacos - $8.99
Chicken Biryani - $9.99
Pepsi - $1.99
Lemonade - $2.49

Order:

Pizza - $8.99
Cheese Loaded Fries - $3.99
Oreo Milkshake - $4.99

Total: $17.97

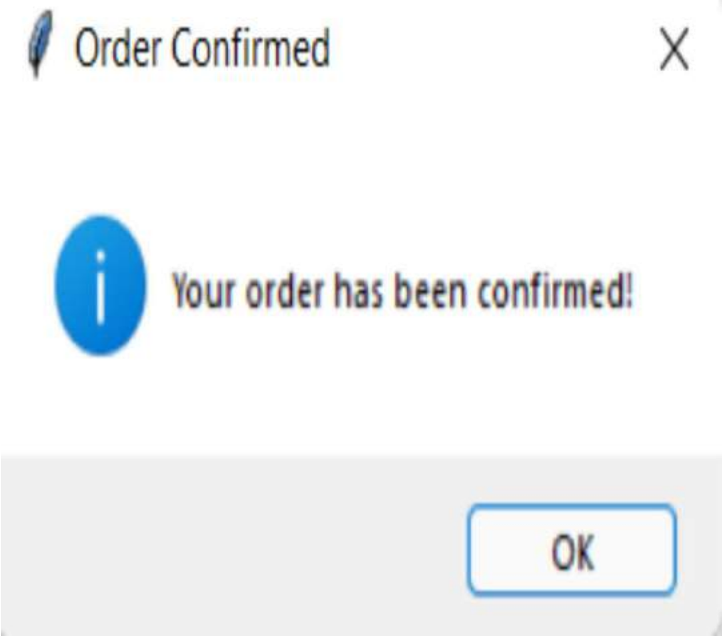
Payment Options:

Customer Name: Suhani

Delivery Address: Woodmardrive 2104 B

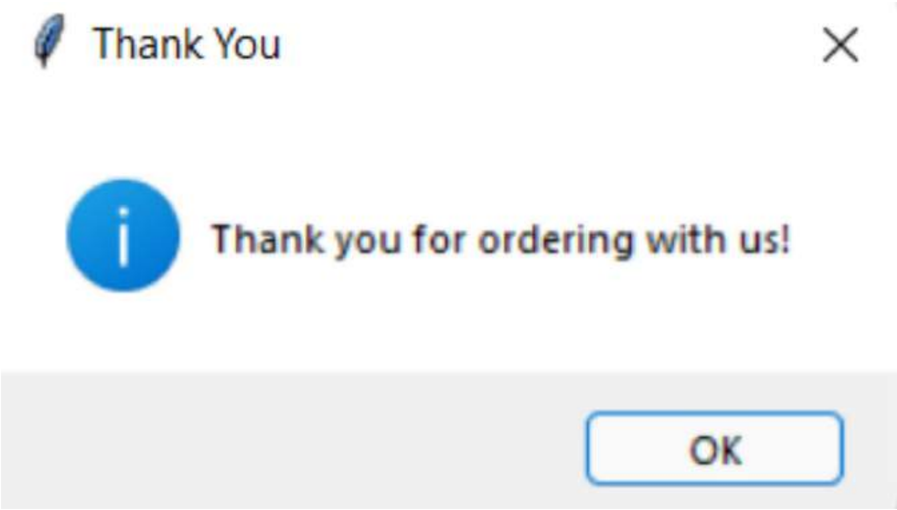
Payment

You have chosen to pay online.
Total amount to pay: $17.97



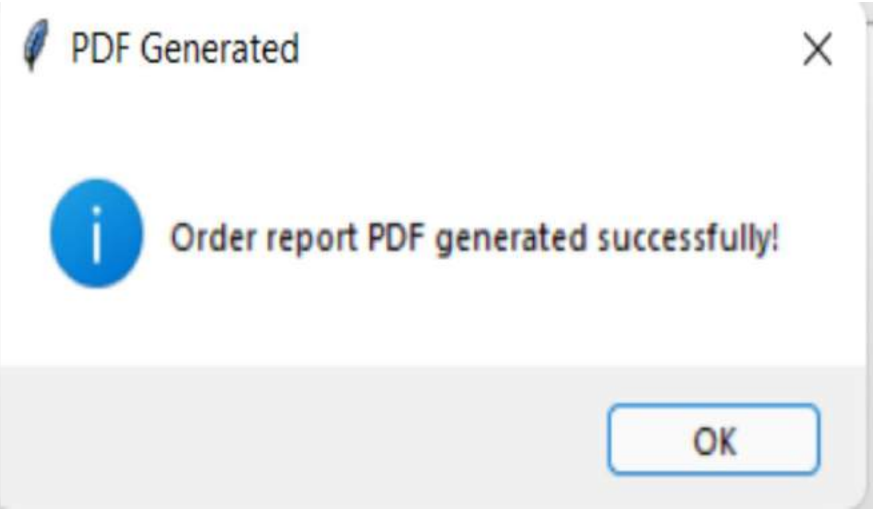
Order Confirmed

Your order has been confirmed!



Thank You

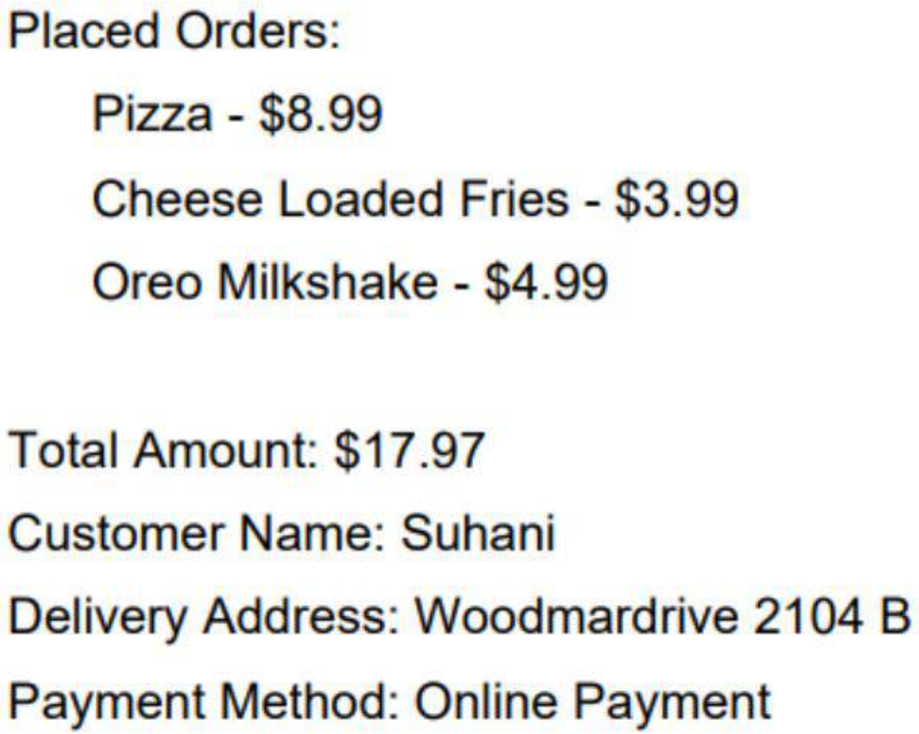
Thank you for ordering with us!



PDF Generated

Order report PDF generated successfully!

ORDER REPORT:



```
Placed Orders:
    Pizza - $8.99
    Cheese Loaded Fries - $3.99
    Oreo Milkshake - $4.99

Total Amount: $17.97
Customer Name: Suhani
Delivery Address: Woodmardrive 2104 B
Payment Method: Online Payment
```

CONCLUSION

- This project is a simple food ordering system that focuses on being easy to understand and use. It relies on Python's tkinter and reportlab libraries for creating menus and generating PDFs. With improvements such as fixing any problems and adding new features like user logins and order history, it can become more helpful for both customers and restaurant owners, making their interactions smoother and safer.