

Code to PDF Conversion

```
import customtkinter as ctk

from tkinter import messagebox, filedialog

from database import Food, Drink, OrderManager, CustomerManager


# Set the theme for customtkinter

ctk.set_appearance_mode("dark")

ctk.set_default_color_theme("blue")


class RestaurantApp:

    def __init__(self, root):

        self.food_manager = Food()

        self.drink_manager = Drink()

        self.order_manager = OrderManager()

        self.customer_manager = CustomerManager()


        # Configure main window

        root.title("Restaurant Management System")

        root.geometry("900x850")

        root.protocol("WM_DELETE_WINDOW", self.on_closing)


        # Center the notebook within the window

        center_frame = ctk.CTkFrame(root)

        center_frame.pack(expand=True, fill="both", padx=50, pady=50)


        # Create notebook (tabbed interface) within the center frame
```

Code to PDF Conversion

```
self.notebook = ctk.CTkTabview(center_frame)

self.notebook.pack(expand=True, fill="both")

# Create tabs

self.menu_tab = self.notebook.add("Add Menu Items")

self.order_tab = self.notebook.add("Place Orders")

self.order_list_tab = self.notebook.add("Show Orders")

self.view_menu_tab = self.notebook.add("View Menu")

self.bill_tab = self.notebook.add("Billing System")

self.customer_tab = self.notebook.add("Manage Customers")
```

```
# Create frames within each tab
```

```
self.create_menu_frame()

self.create_order_frame()

self.create_order_list_frame()

self.create_view_menu_frame()

self.create_bill_frame()

self.create_customer_frame()
```

```
def create_menu_frame(self):
```

```
    frame = ctk.CTkFrame(self.menu_tab, corner_radius=10)

    frame.pack(fill="both", expand=True, padx=10, pady=10)
```

```
# Entry for Multiple Foods and Prices
```

```
    ctk.CTkLabel(frame, text="Enter Foods and Prices (e.g., 'Pizza,10\nBurger,8'):",
```

Code to PDF Conversion

```
anchor="w").pack(pady=10)

self.food_text = ctk.CTkTextbox(frame, height=100, width=400)

self.food_text.pack(pady=10)


# Entry for Multiple Drinks and Prices

        ctk.CTkLabel(frame, text="Enter Drinks and Prices (e.g., 'Cola,3\nWater,1'):",
anchor="w").pack(pady=10)

self.drink_text = ctk.CTkTextbox(frame, height=100, width=400)

self.drink_text.pack(pady=10)


# Add Button

ctk.CTkButton(frame, text="Add Items", command=self.add_items).pack(pady=20)


def create_order_frame(self):

    frame = ctk.CTkFrame(self.order_tab, corner_radius=10)

    frame.pack(fill="both", expand=True, padx=10, pady=10)


# Title

ctk.CTkLabel(frame, text="Place Orders", font=("Arial", 18)).pack(pady=10)


# Food Order Section

food_order_frame = ctk.CTkFrame(frame)

food_order_frame.pack(pady=10)


ctk.CTkLabel(food_order_frame, text="Food ID:", width=15, anchor="e").grid(row=0, column=0,
```

Code to PDF Conversion

```
padx=5, pady=5)

self.food_id = ctk.CTkEntry(food_order_frame, width=150)

self.food_id.grid(row=0, column=1, padx=5, pady=5)


        ctk.CTkLabel(food_order_frame, text="Quantity:", width=15, anchor="e").grid(row=0,
column=2, padx=5, pady=5)

self.food_qty = ctk.CTkEntry(food_order_frame, width=150)

self.food_qty.grid(row=0, column=3, padx=5, pady=5)


# Drink Order Section

drink_order_frame = ctk.CTkFrame(frame)

drink_order_frame.pack(pady=10)


        ctk.CTkLabel(drink_order_frame, text="Drink ID:", width=15, anchor="e").grid(row=0,
column=0, padx=5, pady=5)

self.drink_id = ctk.CTkEntry(drink_order_frame, width=150)

self.drink_id.grid(row=0, column=1, padx=5, pady=5)


        ctk.CTkLabel(drink_order_frame, text="Quantity:", width=15, anchor="e").grid(row=0,
column=2, padx=5, pady=5)

self.drink_qty = ctk.CTkEntry(drink_order_frame, width=150)

self.drink_qty.grid(row=0, column=3, padx=5, pady=5)


# Customer Info Section

ctk.CTkLabel(frame, text="Customer ID (optional):").pack(pady=5)
```

Code to PDF Conversion

```
self.customer_id = ctk.CTkEntry(frame, width=150)
```

```
self.customer_id.pack()
```

```
# Place Order Button
```

```
ctk.CTkButton(frame, text="Place Order", command=self.place_order).pack(pady=20)
```

```
def create_order_list_frame(self):
```

```
    frame = ctk.CTkFrame(self.order_list_tab, corner_radius=10)
```

```
    frame.pack(fill="both", expand=True, padx=10, pady=10)
```

```
    ctk.CTkButton(frame, text="Show Orders", command=self.show_orders).pack(pady=10)
```

```
    self.orders_text = ctk.CTkTextbox(frame, height=200, width=500)
```

```
    self.orders_text.pack()
```

```
def create_view_menu_frame(self):
```

```
    frame = ctk.CTkFrame(self.view_menu_tab, corner_radius=10)
```

```
    frame.pack(fill="both", expand=True, padx=10, pady=10)
```

```
    ctk.CTkButton(frame, text="Show Menu", command=self.show_menu).pack(pady=10)
```

```
    self.menu_text = ctk.CTkTextbox(frame, height=300, width=500)
```

```
    self.menu_text.pack()
```

```
def create_bill_frame(self):
```

```
    frame = ctk.CTkFrame(self.bill_tab, corner_radius=10)
```

```
    frame.pack(fill="both", expand=True, padx=10, pady=10)
```

Code to PDF Conversion

```
ctk.CTkLabel(frame, text="Generated Bill:", font=("Arial", 16)).pack(pady=10)
```

```
self.bill_text = ctk.CTkTextbox(frame, height=200, width=500)
```

```
self.bill_text.pack()
```

```
# Print Button
```

```
ctk.CTkButton(frame, text="Print Bill", command=self.print_bill).pack(pady=20)
```

```
def create_customer_frame(self):
```

```
    frame = ctk.CTkFrame(self.customer_tab, corner_radius=10)
```

```
    frame.pack(fill="both", expand=True, padx=10, pady=10)
```

```
# Customer Registration Form
```

```
ctk.CTkLabel(frame, text="Customer Name:").pack(pady=5)
```

```
self.customer_name = ctk.CTkEntry(frame)
```

```
self.customer_name.pack(pady=5)
```

```
ctk.CTkLabel(frame, text="Customer Contact:").pack(pady=5)
```

```
self.customer_contact = ctk.CTkEntry(frame)
```

```
self.customer_contact.pack(pady=5)
```

```
# Save Customer Button
```

```
ctk.CTkButton(frame, text="Add Customer", command=self.add_customer).pack(pady=10)
```

```
def add_customer(self):
```

Code to PDF Conversion

```
name = self.customer_name.get()
contact = self.customer_contact.get()
self.customer_manager.add_customer(name, contact)
messagebox.showinfo("Customer Added", "Customer successfully added.")
```

Other methods remain the same...

Assume database.py has updated classes for CustomerManager