Ice Cream Haven

CODE DOCUMENTATION & CODE SCREENSHOTS:

```
import tkinter as tk
from tkinter import Label, Entry, Button, Listbox, simpledialog,
messagebox, END, Frame, Canvas, Scrollbar
import sqlite3
from PIL import Image, ImageTk
import os

AMAZON_BLACK = "#0F1111"
AMAZON_ORANGE = "#FF9900"
AMAZON_WHITE = "#FFFFFF"
AMAZON_GRAY = "#D5DBDB"

conn = sqlite3.connect('icecream.db')
cursor = conn.cursor()
cursor.execute('''CREATE TABLE IF NOT EXISTS flavors
                  (name TEXT, seasonal INTEGER, ingredients TEXT, allergens
TEXT)''')
conn.commit()

cursor.execute('''CREATE TABLE IF NOT EXISTS cart
                  (name TEXT, allergens TEXT, user_text TEXT)''')  #
Removed the ALTER TABLE command
conn.commit()

cursor.execute('''CREATE TEMPORARY TABLE IF NOT EXISTS suggested_flavors
                  (name TEXT, seasonal INTEGER, ingredients TEXT, allergens
TEXT)''')
conn.commit()

cursor.execute("DELETE FROM cart")
conn.commit()
cursor.execute("DELETE FROM flavors")
conn.commit()
cursor.execute("DELETE FROM suggested_flavors")
conn.commit()

flavors_to_insert = [
    ("Vanilla", 0, "Milk, Sugar, Vanilla Extract", "Milk"),
    ("Chocolate", 0, "Milk, Sugar, Chocolate", "Milk"),
    ("Strawberry", 1, "Milk, Sugar, Strawberry", "Milk"),
    ("Butterscotch", 0, "Milk, Sugar, Butterscotch Flavoring", "Milk"),
    ("Cookies and Cream", 0, "Milk, Sugar, Chocolate Cookies", "Milk,
Wheat"),
    ("Pista", 0, "Milk, Sugar, Pistachio Nuts", "Milk, Nuts"),
    ("Mango", 1, "Milk, Sugar, Mango Pulp", "Milk")
]

cursor.executemany("INSERT INTO flavors (name, seasonal, ingredients,
allergens) VALUES (?, ?, ?, ?)",
                   flavors_to_insert)
conn.commit()
print("Inserted default flavors into the flavors table")

class IceCreamParlorApp:
```

```
    def __init__(self, root):
        self.root = root
        self.root.title("Ice Cream Haven")
        self.root.geometry("800x600")
        self.root.configure(bg=AMAZON_BLACK)

        self.create_main_window()

    def create_main_window(self):
        self.clear_window()

        self.bg_image = Image.open("5b1omt2a.png")
        self.bg_image = self.bg_image.resize((800, 600), Image.LANCZOS)
        self.bg_photo = ImageTk.PhotoImage(self.bg_image)

        self.background_label = Label(self.root, image=self.bg_photo)
        self.background_label.place(relwidth=1, relheight=1)

        # Overlay frame for content
        self.overlay_frame = Frame(self.root, bg=AMAZON_BLACK, bd=0)
        self.overlay_frame.place(relx=0.5, rely=0.5, anchor='center')

        self.home_label = Label(self.overlay_frame, text="Dive into a World
of Ice Cream Joy!", font=("Arial", 24),
                                fg=AMAZON_WHITE, bg=AMAZON_BLACK)
        self.home_label.grid(row=0, columnspan=2, pady=20)

        # Row for the "View Flavors" and "Suggest a Flavor" buttons
        self.flavors_button = Button(self.overlay_frame, text="View
Flavors", command=self.open_flavors_window,
                                     bg=AMAZON_ORANGE, fg=AMAZON_WHITE,
font=("Arial", 14))
        self.flavors_button.grid(row=1, column=0, padx=10, pady=10)

        self.suggest_button = Button(self.overlay_frame, text="Suggest a
Flavor", command=self.open_suggest_window,
                                     bg=AMAZON_ORANGE, fg=AMAZON_WHITE,
font=("Arial", 14))
        self.suggest_button.grid(row=1, column=1, padx=10, pady=10)

        self.cart_button = Button(self.overlay_frame, text="View Cart",
command=self.open_cart_window, bg=AMAZON_ORANGE,
                                  fg=AMAZON_WHITE, font=("Arial", 14))
        self.cart_button.grid(row=2, columnspan=2, pady=10)

    def clear_window(self):
        for widget in self.root.winfo_children():
            widget.destroy()

    def open_flavors_window(self):
        self.clear_window()

        self.frame = Frame(self.root, bg=AMAZON_BLACK)
        self.frame.pack(expand=True, fill="both")
        self.frame.columnconfigure(0, weight=1)

        self.scrollbar = Scrollbar(self.frame, orient="vertical")
        self.canvas = Canvas(self.frame, bg=AMAZON_BLACK,
yscrollcommand=self.scrollbar.set)
        self.scrollbar.config(command=self.canvas.yview)
```

```
        self.scrollbar.pack(side="right", fill="y")
        self.canvas.pack(side="left", fill="both", expand=True)

        self.inner_frame = Frame(self.canvas, bg=AMAZON_BLACK)
        self.canvas.create_window((0, 0), window=self.inner_frame,
anchor="nw")

        self.load_flavors()

        self.search_label = Label(self.inner_frame, text="Search Flavors:",
font=("Arial", 12), fg=AMAZON_WHITE,
                                  bg=AMAZON_BLACK)
        self.search_label.grid(row=0, column=0, pady=(10, 5), sticky="w")

        self.search_entry = Entry(self.inner_frame, width=30)
        self.search_entry.grid(row=0, column=1, pady=(10, 5), padx=(5, 0))
        # Bind the <Return> event to call the search_flavors method
        self.search_entry.bind('<Return>', lambda event:
self.search_flavors())

        self.search_button = Button(self.inner_frame, text="Search",
command=self.search_flavors, bg=AMAZON_ORANGE,
                                    fg=AMAZON_WHITE, font=("Arial", 12))
        self.search_button.grid(row=0, column=2, pady=(10, 5), padx=(5, 0))

        self.back_button = Button(self.inner_frame, text="Back",
command=self.create_main_window, bg=AMAZON_ORANGE,
                                  fg=AMAZON_WHITE, font=("Arial", 12))
        self.back_button.grid(row=0, column=3, pady=(10, 5), padx=(5, 0))

        self.canvas.bind("<Configure>", self.on_canvas_configure)

    def on_canvas_configure(self, event):
        self.canvas.configure(scrollregion=self.canvas.bbox("all"))

    def load_flavors(self):
        cursor.execute("SELECT name FROM flavors")
        flavors = cursor.fetchall()

        row_index = 1
        col_index = 0

        for flavor in flavors:
            if col_index == 5:
                row_index += 1
                col_index = 0

            self.create_flavor_button(flavor[0], row_index, col_index)
            col_index += 1

        cursor.execute("SELECT name FROM suggested_flavors")
        suggested_flavors = cursor.fetchall()

        for flavor in suggested_flavors:
            if col_index == 5:
                row_index += 1
                col_index = 0

            self.create_flavor_button(flavor[0], row_index, col_index)
            col_index += 1
```

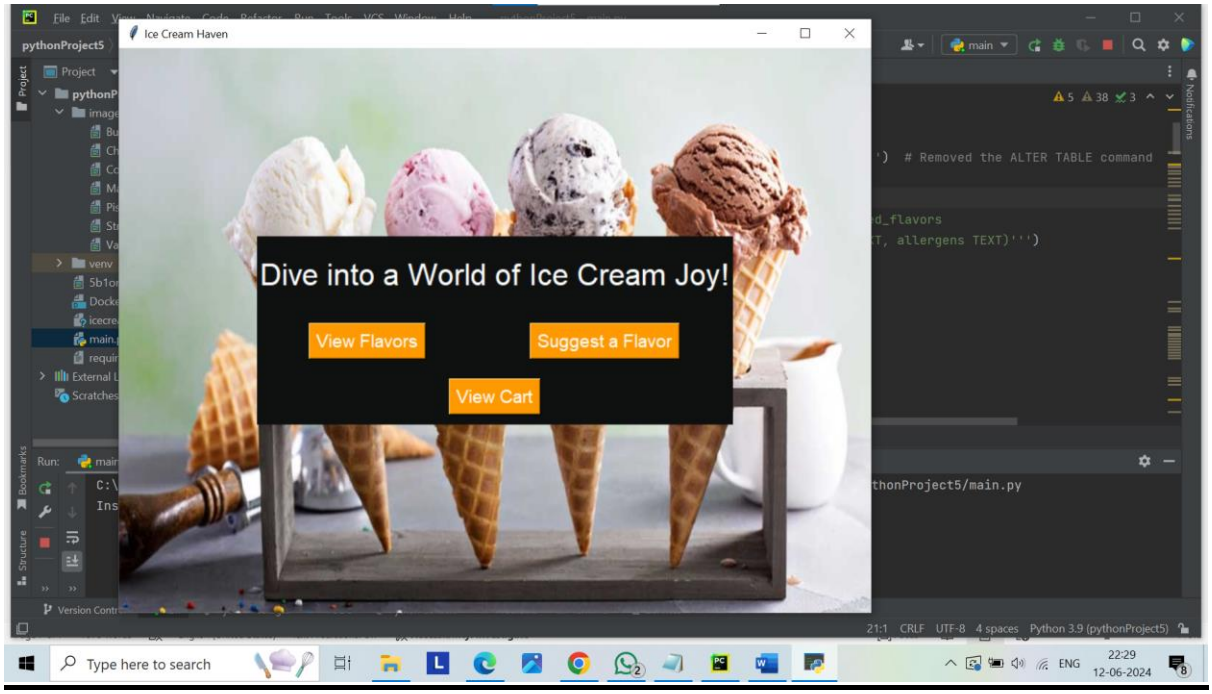
```
    def create_flavor_button(self, flavor_name, row, column):
        image_path =
os.path.join("C:/Users/shanm/PycharmProjects/pythonProject5/images",
f"{flavor_name}.jpg")
        if os.path.exists(image_path):
            image = Image.open(image_path)
            image = image.resize((100, 100), Image.LANCZOS)

            photo = ImageTk.PhotoImage(image)
            button = Button(self.inner_frame, text=flavor_name,
image=photo, compound="top", bg=AMAZON_ORANGE,
                            fg=AMAZON_WHITE, font=("Arial", 12),
                            command=lambda name=flavor_name:
self.add_to_cart(name))
            button.image = photo  # Keep a reference to avoid garbage
collection
            button.grid(row=row, column=column, padx=10, pady=10)
        else:
            button = Button(self.inner_frame, text=flavor_name,
bg=AMAZON_ORANGE, fg=AMAZON_WHITE, font=("Arial", 12),
                            command=lambda name=flavor_name:
self.add_to_cart(name))
            button.grid(row=row, column=column, padx=10, pady=10)

    def search_flavors(self):
        query = self.search_entry.get().lower()

        # Clear existing flavor buttons
        for widget in self.inner_frame.winfo_children():
            if isinstance(widget, Button) and widget.cget("text") !=
"Back":
                widget.destroy()

        # Search for matching flavors
        cursor.execute("SELECT name FROM flavors WHERE LOWER(name) LIKE ?",
('%' + query + '%',))
        flavors = cursor.fetchall()
        row_index = 1
        col_index = 0
        for flavor in flavors:
            if col_index == 5:
                row_index += 1
                col_index = 0

            self.create_flavor_button(flavor[0], row_index, col_index)
            col_index += 1

        # Search for suggested flavors
        cursor.execute("SELECT name FROM suggested_flavors WHERE
LOWER(name) LIKE ?", ('%' + query + '%',))
        suggested_flavors = cursor.fetchall()
        for flavor in suggested_flavors:
            if col_index == 5:
                row_index += 1
                col_index = 0

            self.create_flavor_button(flavor[0], row_index, col_index)
            col_index += 1

        # Check if the back button already exists and create it if not
        if not hasattr(self, 'back_button') or not
```

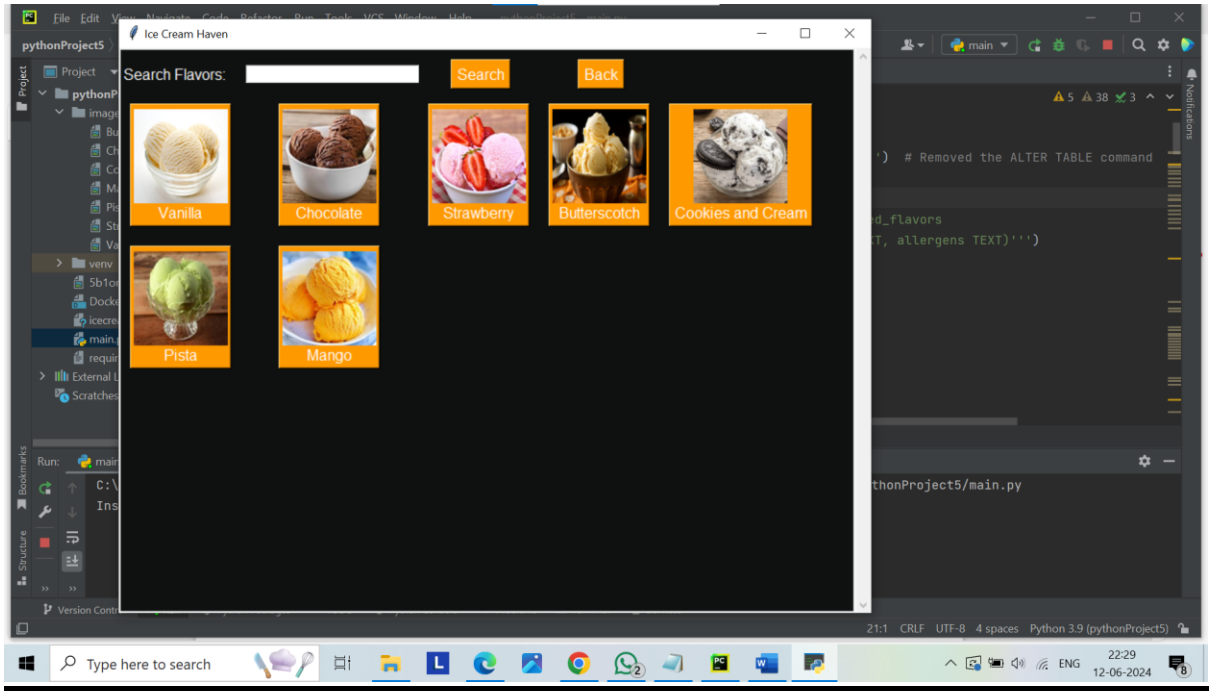
```
self.back_button.winfo_exists():
            self.back_button = Button(self.inner_frame, text="Back",
command=self.create_main_window, bg=AMAZON_ORANGE,
                                      fg=AMAZON_WHITE, font=("Arial", 14))
            self.back_button.grid(row=row_index + 1, column=0, pady=10)

    # Assuming `self.create_main_window` will re-create the back button,
    # make sure to clean up or update any reference to self.back_button
accordingly in that method.

    def add_to_cart(self, flavor_name):
        allergens = cursor.execute("SELECT allergens FROM flavors WHERE
name = ?", (flavor_name,)).fetchone()
        if not allergens:
            allergens = cursor.execute("SELECT allergens FROM
suggested_flavors WHERE name = ?", (flavor_name,)).fetchone()

        if allergens:
            allergens = allergens[0]
        else:
            allergens = "Unknown"

        user_text = simpledialog.askstring("Add to Cart", f"Add a note for
{flavor_name} (optional):")

        cursor.execute("INSERT INTO cart (name, allergens, user_text)
VALUES (?, ?, ?)",
                       (flavor_name, allergens, user_text))
        conn.commit()

        messagebox.showinfo("Added to Cart", f"{flavor_name} has been added
to your cart.")

    def open_suggest_window(self):
        self.clear_window()

        self.suggest_frame = Frame(self.root, bg=AMAZON_BLACK)
        self.suggest_frame.pack(expand=True, fill="both")

        self.suggest_label = Label(self.suggest_frame, text="Suggest a New
Flavor", font=("Arial", 18), fg=AMAZON_WHITE,
                                   bg=AMAZON_BLACK)
        self.suggest_label.pack(pady=20)

        self.name_label = Label(self.suggest_frame, text="Name:",
font=("Arial", 14), fg=AMAZON_WHITE, bg=AMAZON_BLACK)
        self.name_label.pack(pady=5)
        self.name_entry = Entry(self.suggest_frame, font=("Arial", 14))
        self.name_entry.pack(pady=5)

        self.seasonal_label = Label(self.suggest_frame, text="Seasonal (0
or 1):", font=("Arial", 14), fg=AMAZON_WHITE,
                                    bg=AMAZON_BLACK)
        self.seasonal_label.pack(pady=5)
        self.seasonal_entry = Entry(self.suggest_frame, font=("Arial", 14))
        self.seasonal_entry.pack(pady=5)

        self.ingredients_label = Label(self.suggest_frame,
text="Ingredients:", font=("Arial", 14), fg=AMAZON_WHITE,
                                       bg=AMAZON_BLACK)
        self.ingredients_label.pack(pady=5)
```

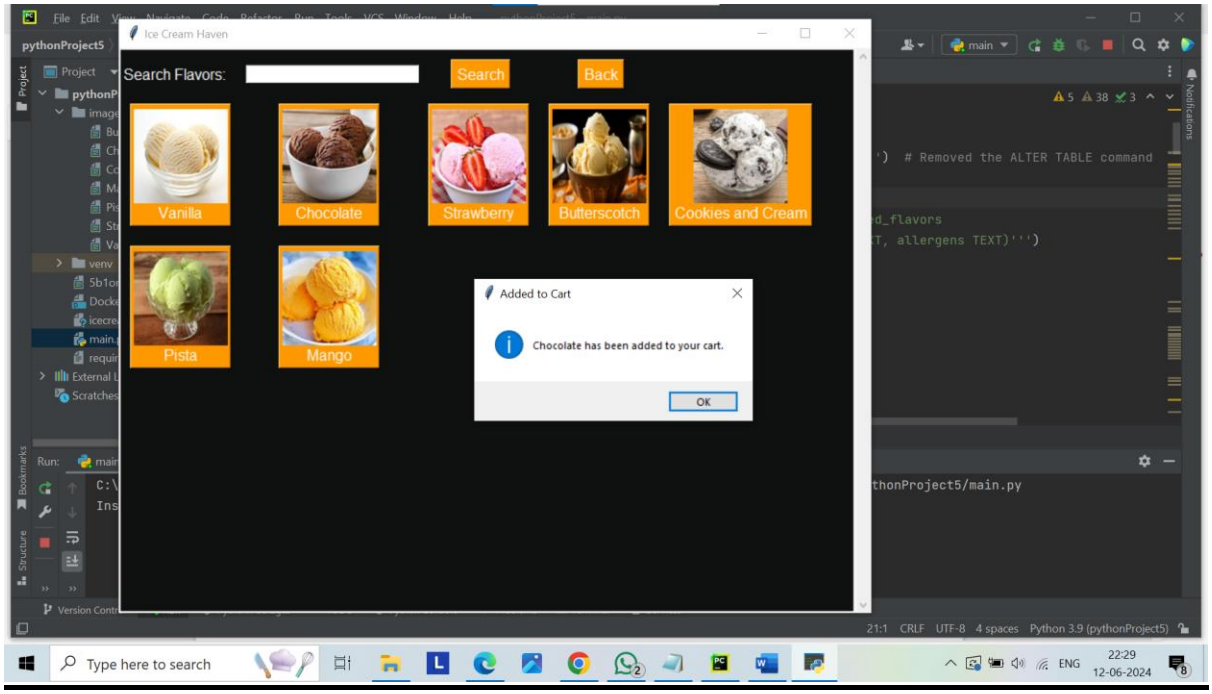
```
        self.ingredients_entry = Entry(self.suggest_frame, font=("Arial",
14))
        self.ingredients_entry.pack(pady=5)

        self.allergens_label = Label(self.suggest_frame, text="Allergens:",
font=("Arial", 14), fg=AMAZON_WHITE,
                                     bg=AMAZON_BLACK)
        self.allergens_label.pack(pady=5)
        self.allergens_entry = Entry(self.suggest_frame, font=("Arial",
14))
        self.allergens_entry.pack(pady=5)

        self.submit_button = Button(self.suggest_frame, text="Submit",
command=self.submit_suggestion, bg=AMAZON_ORANGE,
                                    fg=AMAZON_WHITE, font=("Arial", 14))
        self.submit_button.pack(pady=20)

        self.back_button = Button(self.suggest_frame, text="Back",
command=self.create_main_window, bg=AMAZON_ORANGE,
                                  fg=AMAZON_WHITE, font=("Arial", 14))
        self.back_button.pack()

    def submit_suggestion(self):
        name = self.name_entry.get()
        seasonal = self.seasonal_entry.get()
        ingredients = self.ingredients_entry.get()
        allergens = self.allergens_entry.get()

        cursor.execute("INSERT INTO suggested_flavors (name, seasonal,
ingredients, allergens) VALUES (?, ?, ?, ?)",
                       (name, seasonal, ingredients, allergens))
        conn.commit()

        messagebox.showinfo("Suggestion Submitted", f"Your suggestion for
{name} has been submitted!")

        self.create_main_window()

    def open_cart_window(self):
        self.clear_window()

        self.cart_frame = Frame(self.root, bg=AMAZON_BLACK)
        self.cart_frame.pack(expand=True, fill="both")

        self.cart_label = Label(self.cart_frame, text="Your Cart",
font=("Arial", 18), fg=AMAZON_WHITE, bg=AMAZON_BLACK)
        self.cart_label.pack(pady=20)

        # Frame to hold the cart items
        self.items_frame = Frame(self.cart_frame, bg=AMAZON_BLACK)
        self.items_frame.pack(pady=10)

        cursor.execute("SELECT name, user_text FROM cart")
        cart_items = cursor.fetchall()

        for item in cart_items:
            name, user_text = item
            display_text = f"{name}"
            if user_text:
                display_text += f" - {user_text}"
```

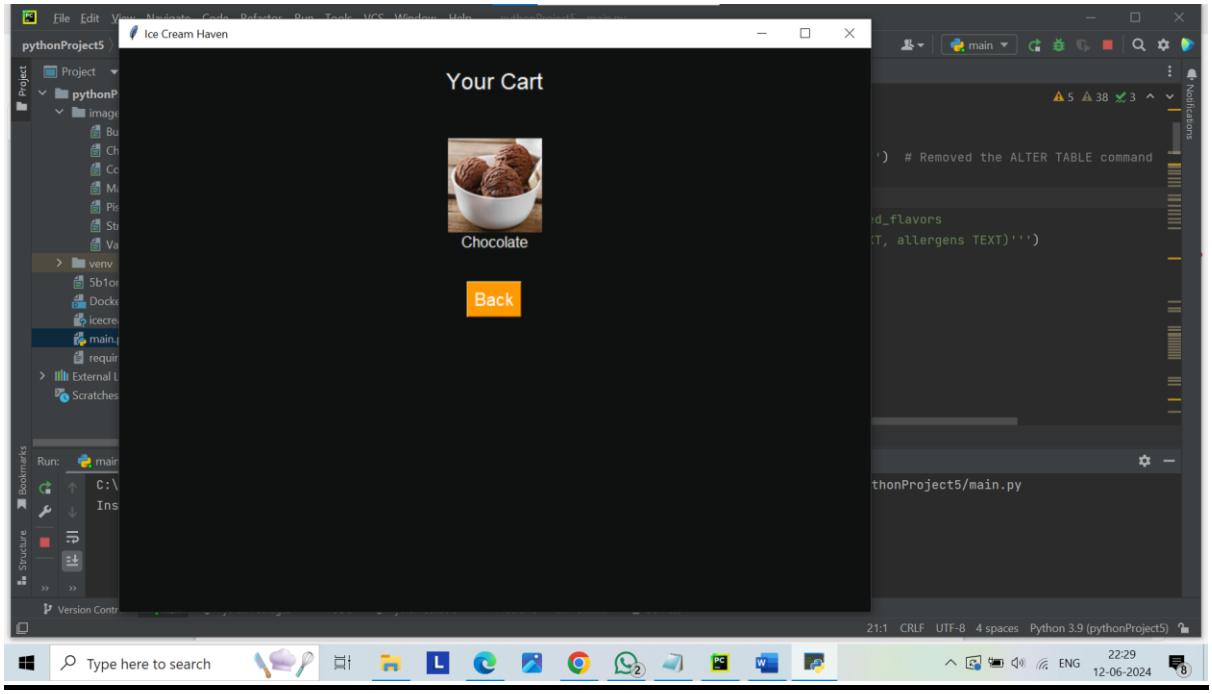
```
            # Load the image if it exists
            image_path =
os.path.join("C:/Users/shanm/PycharmProjects/pythonProject5/images",
f"{name}.jpg")
            if os.path.exists(image_path):
                image = Image.open(image_path)
                image = image.resize((100, 100), Image.LANCZOS)
                photo = ImageTk.PhotoImage(image)

                item_frame = Frame(self.items_frame, bg=AMAZON_BLACK)
                item_frame.pack(pady=10)

                label = Label(item_frame, text=display_text, image=photo,
compound="top", bg=AMAZON_BLACK,
                              fg=AMAZON_WHITE, font=("Arial", 12))
                label.image = photo  # Keep a reference to avoid garbage
collection
                label.pack(side="left", padx=10)
            else:
                label = Label(self.items_frame, text=display_text,
bg=AMAZON_BLACK, fg=AMAZON_WHITE, font=("Arial", 12))
                label.pack(pady=10)

        self.back_button = Button(self.cart_frame, text="Back",
command=self.create_main_window, bg=AMAZON_ORANGE,
                                  fg=AMAZON_WHITE, font=("Arial", 14))
        self.back_button.pack(pady=10)


if __name__ == "__main__":
    root = tk.Tk()
    app = IceCreamParlorApp(root)
    root.mainloop()
```

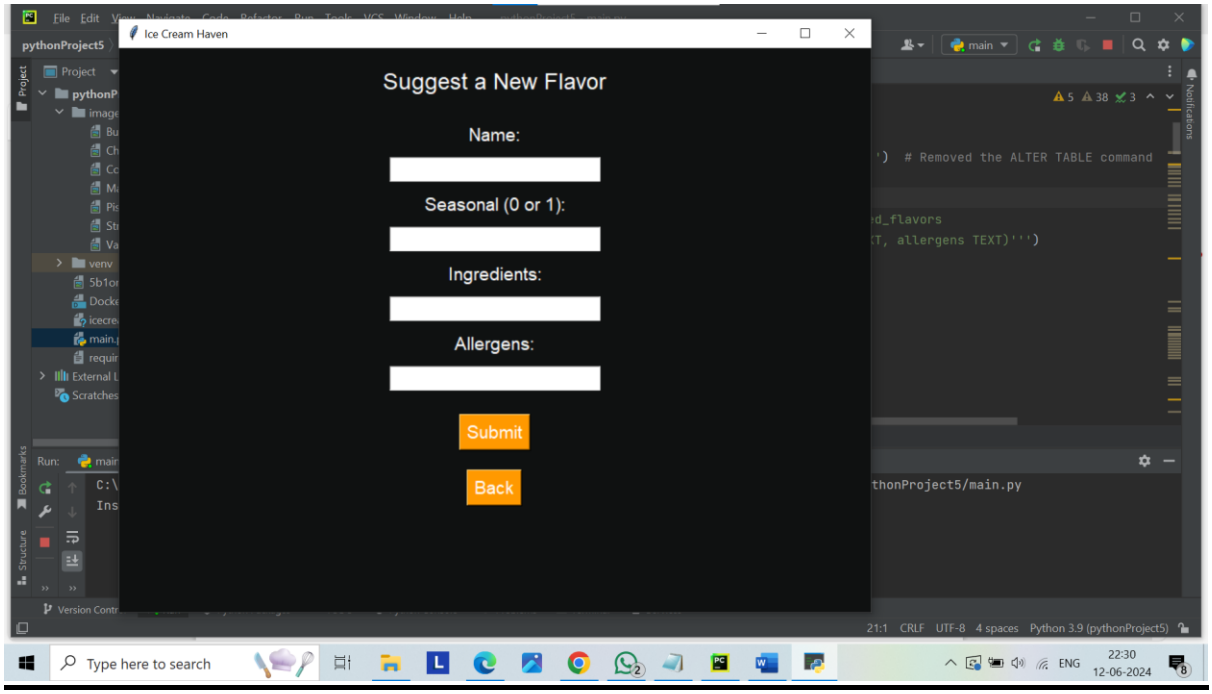
CODE SCREENSHOTS:


Ice Cream Haven
Dive into a World of Ice Cream Joy!
View Flavors
Suggest a Flavor
View Cart


Ice Cream Haven
Search Flavors:
Search
Back
Vanilla
Chocolate
Strawberry
Butterscotch
Cookies and Cream
Pista
Mango


Ice Cream Haven
Search Flavors:
Search
Back
Vanilla
Chocolate
Strawberry
Butterscotch
Cookies and Cream
Pista
Mango
Added to Cart
Chocolate has been added to your cart.
OK


Ice Cream Haven
Your Cart
Chocolate
Back


Ice Cream Haven
Suggest a New Flavor
Name:
Seasonal (0 or 1):
Ingredients:
Allergens:
Submit
Back
pythonProject5
Type here to search
ENG
22:30
12-06-2024