```
# Name: Trien Bang Huynh
# Lab 3: Web scraping and data storage with requests, beautifulsoup, sqlite3,
review tkinter.
# lab3front.py: DisplayWin class, DialogWin class, MainWin class

import tkinter as tk
import tkinter.messagebox as tkmb
import sqlite3
import webbrowser


class DisplayWin(tk.Toplevel):
    '''
    A class which shows info of selected restaurants
    '''
    def __init__(self, master, restaurantID, connDB):
        super().__init__(master)
        self.transient(master)

        # Connect to database
        self.conn = connDB
        self.cur = self.conn.cursor()

        self.cur.execute("SELECT * FROM Main WHERE Main.id = ?",(restaurantID,))
        restaurant = self.cur.fetchone()

        name, address  = restaurant[1], restaurant[6]

        self.cur.execute("SELECT Costs.cost FROM Main JOIN Costs ON Main.cost =
Costs.id AND Main.id = ?",(restaurantID,))
        cost = self.cur.fetchone()[0]

        self.cur.execute("SELECT Cuisine.cuisine FROM Main JOIN Cuisine ON
Main.kind = Cuisine.id AND Main.id = ?",(restaurantID,))
        cuisine = self.cur.fetchone()[0]

        url = restaurant[2]


        tk.Label(self, text= name, font=('Times', 15), fg="blue").pack(padx=15,
pady=10)
        tk.Label(self, text= address, font=('Times', 15)).pack(padx=15, pady=10)
        tk.Label(self, text= f"Cost: {cost}", font=('Times', 15)).pack(padx=15,
pady=10)
        tk.Label(self, text= f"Cuisine: {cuisine}", font=('Times',
15),fg="blue").pack(padx=15, pady=10)
        tk.Button(self, text="Visit Webpage", font=('Times', 15), fg="blue",
command = lambda: webbrowser.open_new(url)).pack(padx=15, pady=10)




class DialogWin(tk.Toplevel):
    '''
    A class which interact and get input from user
    '''
    def __init__(self, master, connDB):
        super().__init__(master)
        self.grab_set()
```

```
        self.focus_set()
        self.transient(master)
        self.protocol("WM_DELETE_WINDOW", self.closeWin)
        self._selection = ()
        self.conn = connDB


    def displayCity(self):
        '''
        A method which display a list of cities for user to select
        '''
        tk.Label(self, text="Click on a city to select", font=('Times',
15)).grid( row=0,padx=15, pady=10)

        # Listbox and Scrollbar
        self.listbox = tk.Listbox(self, height=6)
        self.scrollbar = tk.Scrollbar(self, orient=tk.VERTICAL,
command=self.listbox.yview)
        self.listbox.configure(yscrollcommand=self.scrollbar.set)

        # Connect to database
        self.cur = self.conn.cursor()

        # add items to the listbox
        for city in self.cur.execute("SELECT * FROM Locations") :
            self.listbox.insert(tk.END,city[1])  # city[0] is city's ID


        self.listbox.grid(row=1, column=0,ipadx=5, padx=20, pady=20, sticky="nsew")
        self.scrollbar.grid(row=1, column=1, sticky="ns")

        # Select button
        tk.Button(self, text="Click to select", font=('Times', 15), command=
self.onClicked).grid(row=2, column=0, columnspan=2, padx=20, pady=20)

    def displayCuisine(self):
        '''
        A method which display a list of cuisine for user to select
        '''
        tk.Label(self, text="Click on a cuisine to select", font=('Times',
15)).grid( row=0,padx=15, pady=10)

        # Listbox and Scrollbar
        self.listbox = tk.Listbox(self, height=6)
        self.scrollbar = tk.Scrollbar(self, orient=tk.VERTICAL,
command=self.listbox.yview)
        self.listbox.configure(yscrollcommand=self.scrollbar.set)

        # Connect to database

        self.cur = self.conn.cursor()

        # add items to the listbox
        for city in self.cur.execute("SELECT * FROM Cuisine") :
            self.listbox.insert(tk.END,city[1])  # cuisine[0] is cuisine's ID



        self.listbox.grid(row=1, column=0,ipadx=5, padx=20, pady=20, sticky="nsew")
```

```
        self.scrollbar.grid(row=1, column=1, sticky="ns")


        # Select button
        tk.Button(self, text="Click to select", font=('Times', 15), command=
self.onClicked).grid(row=2, column=0, columnspan=2, padx=20, pady=20)


    def retrieveCityRestaurant(self, cityID):
        '''
        A method which retrieve restaurant from given cityID
        '''
        tk.Label(self, text="Click on a restaurant to select", font=('Times',
15)).grid(row=0,padx=15, pady=10)

        # Listbox and Scrollbar
        self.listbox = tk.Listbox(self, height=6, selectmode='multiple')
        self.scrollbar = tk.Scrollbar(self, orient=tk.VERTICAL,
command=self.listbox.yview)
        self.listbox.configure(yscrollcommand=self.scrollbar.set)

        # Connect to database

        self.cur = self.conn.cursor()

        self.cur.execute("SELECT Main.name FROM Main WHERE Main.loc = ?",(cityID,))

        for restaurant in self.cur.fetchall() :
            self.listbox.insert(tk.END,restaurant[0])



        self.listbox.grid(row=1, column=0,ipadx=5, padx=20, pady=20, sticky="nsew")
        self.scrollbar.grid(row=1, column=1, sticky="ns")

         # Select button
        tk.Button(self, text="Click to select", font=('Times', 15), command=
self.onClicked).grid(row=2, column=0, columnspan=2, padx=20, pady=20)


    def retrieveCuisineRestaurant(self, cuisineID):
        '''
        A method which retrieve restaurant from given cuisineID
        '''
        tk.Label(self, text="Click on a restaurant to select", font=('Times',
15)).grid(row=0,padx=15, pady=10)

        # Listbox and Scrollbar
        self.listbox = tk.Listbox(self, height=6, selectmode='multiple')
        self.scrollbar = tk.Scrollbar(self, orient=tk.VERTICAL,
command=self.listbox.yview)
        self.listbox.configure(yscrollcommand=self.scrollbar.set)

        # Connect to database

        self.cur = self.conn.cursor()

        self.cur.execute("SELECT Main.name FROM Main WHERE Main.kind = ?",
(cuisineID,))
```

```
        for restaurant in self.cur.fetchall() :
            self.listbox.insert(tk.END,restaurant[0])


        self.listbox.grid(row=1, column=0,ipadx=5, padx=20, pady=20, sticky="nsew")
        self.scrollbar.grid(row=1, column=1, sticky="ns")

         # Select button
        tk.Button(self, text="Click to select", font=('Times', 15), command=
self.onClicked).grid(row=2, column=0, columnspan=2, padx=20, pady=20)


    def onClicked(self):
        self._selection = self.listbox.curselection()
        self.closeWin()

    @property
    def getSelection(self):
        return self._selection

    def closeWin(self):
        self.destroy()


class MainWin(tk.Tk):
    '''
    A class which asks user to find restaurants based on city or cuisine
    '''
    def __init__(self):
        super().__init__()
        self.title("Restaurants")
        tk.Label(self, text="Local Michelin Guild Restaurants",
fg="green",font=('Times', 17)).grid(row=0, column=0, columnspan=3, pady=10,
padx=10)
        tk.Label(self, text="Search by", fg="black",font=('Times', 15)).grid(row=1,
column=0, columnspan=3, pady=10)
        tk.Button(self, text="City", fg="blue", command=
self.processCity).grid(row=2, column=0, padx=15, pady=10)
        tk.Button(self, text="Cuisine", fg="blue",command=
self.processCuisine).grid(row=2, column=1, padx=15, pady=10)

        self.protocol("WM_DELETE_WINDOW", self.closeWin)

        # Connect to database
        try:
            self.conn = sqlite3.connect('data.db')
            self.cur = self.conn.cursor()
        except sqlite3.Error:
            tkmb.showerror("Error", "Failed to open database")
            self.destroy()
            self.quit()


    def processCity(self):
        '''
        A method which displays a list of cities from database, process user's
choice and call DisplayWin
        '''
```

```
        self.dialogWin1 = DialogWin(self,self.conn)
        self.dialogWin1.displayCity()
        self.wait_window(self.dialogWin1)

       # self.dialog_window.getSelection returns a tuple of indices of choices in
the listbox
        if len(self.dialogWin1.getSelection) != 0:

            # indices of choices in the listbox starts with 0, so should be +1 to
get correct ID
            cityID  = self.dialogWin1.getSelection[0] + 1

            # create a new dialog win to retrieve restaurant's city
            self.dialogWin2 = DialogWin(self,self.conn)
            self.dialogWin2.retrieveCityRestaurant(cityID)
            self.wait_window(self.dialogWin2)

            # fetch restaurants' ID that have the same cityID
            self.cur.execute("SELECT Main.id FROM Main WHERE Main.loc = ?",
(cityID,))
            selectedRestaurant = self.cur.fetchall()

            for i in self.dialogWin2.getSelection:
                DisplayWin(self,selectedRestaurant[i][0],self.conn) # pass the
restaurant's ID with indices of selected restaurant from the dialog to DisplayWin



    def processCuisine(self):
        '''
        A method which displays a list of cuisine from database, process user's
choice and call DisplayWin
        '''
        self.dialogWin1 = DialogWin(self,self.conn)
        self.dialogWin1.displayCuisine()
        self.wait_window(self.dialogWin1)

        if len(self.dialogWin1.getSelection) != 0:

            # indices of choices in the listbox starts with 0, so should be +1 to
get correct ID
            cuisineID  = self.dialogWin1.getSelection[0] + 1

            # create a new dialog win to retrieve restaurant's city
            self.dialogWin2 = DialogWin(self,self.conn)
            self.dialogWin2.retrieveCuisineRestaurant(cuisineID)
            self.wait_window(self.dialogWin2)

            # fetch restaurants' ID that have the same cuisineID
            self.cur.execute("SELECT Main.id FROM Main WHERE Main.kind = ?",
(cuisineID,))
            selectedRestaurant = self.cur.fetchall()

            for i in self.dialogWin2.getSelection:
                DisplayWin(self,selectedRestaurant[i][0], self.conn) # pass the
restaurant's ID that was selected from the dialog to DisplayWin
```

```
    def closeWin(self):
        self.conn.close()
        self.destroy()
        self.quit()

app = MainWin()
app.mainloop()
```