from tkinter import \*  
import tkinter as tk  
import datetime  
import sqlite3  
from tkinter import messagebox  
  
  
def login():  
 user = username.get()  
 code = password.get()  
 # Creating seccond window  
 if user == '' and code == '':  
 messagebox.showerror('Invalid', 'Please enter Username and Password')  
 elif user == '':  
 messagebox.showerror('Invalid', 'Username is required')  
 elif code == '':  
 messagebox.showerror('Invalid', 'Password field required')  
 else:  
 # Connecting to the database  
 connection = sqlite3.connect('users.sqlite')  
 cursor = connection.cursor()  
  
 # Checking if the username and password exist in the database  
 cursor.execute('SELECT \* FROM users WHERE username=? AND password=?', (user, code))  
 user\_exists = cursor.fetchone()  
  
 # Closing the connection  
  
 # If the username and password exist, then the user is logged in  
 if user\_exists:  
 select\_recipe()  
 else:  
 messagebox.showerror('Invalid', 'Invalid username or Password')  
  
  
def create\_account():  
 user = username.get()  
 code = password.get()  
 # Creating seccond window  
 if user != '' and code != '':  
 # Connecting to the database  
 connection = sqlite3.connect('users.sqlite')  
 cursor = connection.cursor()  
  
 # Creating a new user  
 cursor.execute('INSERT INTO users (username, password) VALUES (?, ?)', (user, code))  
 connection.commit()  
  
 # Closing the connection  
 connection.close()  
  
 # Creating a success message  
 messagebox.showinfo('Success', 'Account created successfully!')  
 else:  
 messagebox.showerror('Error', 'Please enter a username and password')  
  
  
def main\_screen():  
 # Creation of the login screen/window  
  
 global screen  
 global username  
 global password  
  
 screen = Tk()  
 screen.geometry('1280x720+150+80')  
 screen.configure(bg='grey')  
  
 # icon  
 screen.title('Login')  
  
 # Label title  
 lblTitle = Label(text='Login System', font=('arial', 50, 'bold'), fg='black', bg='grey')  
 lblTitle.pack(pady=50)  
 # Border for username and password  
 bordercolor = Frame(screen, bg='black', width=800, height=400)  
 bordercolor.pack()  
  
 mainframe = Frame(bordercolor, bg='grey', width=800, height=400)  
 mainframe.pack(padx=20, pady=20)  
  
 Label(mainframe, text='Username', font=('arial', 30, 'bold'), bg='grey').place(x=100, y=50)  
 Label(mainframe, text='Password', font=('arial', 30, 'bold'), bg='grey').place(x=101, y=150)  
  
 username = StringVar()  
 password = StringVar()  
 # Entry boxes for username and password  
 entry\_username = Entry(mainframe, textvariable=username, width=12, bd=2, font=('arial', 30))  
 entry\_username.place(x=400, y=50)  
 entry\_password = Entry(mainframe, textvariable=password, width=12, bd=2, font=('arial', 30), show='\*')  
 entry\_password.place(x=400, y=150)  
  
 # Defining reset command  
  
 def reset():  
 entry\_username.delete(0, END)  
 entry\_password.delete(0, END)  
  
 # Login, Reset and Exit buttons  
  
 Button(mainframe, text='Login', height='2', width=23, bg='#ed3833', fg='white', bd=0, command=login).place(x=100,  
 y=250)  
 Button(mainframe, text='Reset', height='2', width=23, bg='#1089ff', fg='white', bd=0, command=reset).place(x=300,  
 y=250)  
 Button(mainframe, text='Create Account', height='2', width=23, bg='#00bd56', fg='white', bd=0,  
 command=create\_account).place(x=500, y=250)  
  
 screen.mainloop()  
  
  
def select\_recipe():  
 global recipeList  
 global listbox  
 global recipe  
 connection = sqlite3.connect('recipes.sqlite')  
 cursor = connection.cursor()  
  
 # Retrieving the recipes from the database  
 cursor.execute('SELECT \* FROM recipes')  
 recipes = cursor.fetchall()  
  
 # Closing the connection  
 connection.close()  
  
 # Creating a new window to display the recipe  
 recipeList = Toplevel(screen)  
 recipe = "Recipes"  
 recipeList.title(recipe)  
 recipeList.geometry('1024x768')  
 recipeList.configure(bg='#318AE4')  
 recipeList.resizable(False, False)  
  
 # Creating a listbox to display the recipes  
 listbox = Listbox(recipeList, width=40, height=20)  
 listbox.pack()  
 listbox.bind('<<ListboxSelect>>', lambda event: show\_recipe())  
  
 # Adding the recipes to the listbox  
 for recipe in recipes:  
 listbox.insert(END, recipe[0])  
  
 # Creating a button to close the window  
 button\_close = Button(recipeList, text='Close Window', width=11, height=2, borderwidth=2,  
 command=recipeList.quit)  
 button\_close.pack()  
  
 #Create Recipe Button  
 recipe\_button = Button(recipeList, text="Create Recipe", width=11, height=2, command=create\_recipe)  
 recipe\_button.pack()  
  
 #Delete Recipe Button  
 delete\_button = Button(recipeList, text="Delete Recipe", width=11, height=2, command=delete\_recipe)  
 delete\_button.pack()  
  
  
def show\_recipe():  
 global selected\_recipe  
 # Get the selected recipe from the listbox  
 selected\_recipe = listbox.get(listbox.curselection())  
  
 # Open a new window to show the recipe  
 recipe\_window = Toplevel(recipeList)  
 recipe\_window.title(selected\_recipe)  
 recipe\_window.geometry('1024x768')  
 recipe\_window.configure(bg='#318AE4')  
 recipe\_window.resizable(True, True)  
  
 #Connect to database  
 connection = sqlite3.connect('recipes.sqlite')  
 cursor = connection.cursor()  
  
 #Get selected recipe  
 cursor.execute('SELECT \* FROM recipes WHERE name=?', (selected\_recipe,))  
 recipe = cursor.fetchone()  
 connection.close()  
  
 #List out selected recipe  
 Label(recipe\_window, text="Name:").grid(row=0, column=0, sticky=W)  
 Label(recipe\_window, text=recipe[0]).grid(row=0, column=1, sticky=W)  
 Label(recipe\_window, text="Ingredients:").grid(row=1, column=0, sticky=W)  
 Label(recipe\_window, text=recipe[1]).grid(row=1, column=1, sticky=W)  
 Label(recipe\_window, text="Instructions:").grid(row=2, column=0, sticky=W)  
 Label(recipe\_window, text=recipe[2]).grid(row=2, column=1, sticky=W)  
  
  
def delete\_recipe():  
 #Delete selected recipe  
 selected\_recipe = listbox.get(listbox.curselection())  
 connection = sqlite3.connect('recipes.sqlite')  
 cursor = connection.cursor()  
 cursor.execute('DELETE FROM recipes WHERE name=?', (selected\_recipe,))  
 connection.commit()  
 connection.close()  
 messagebox.showinfo("Recipe Deleted", "Recipe has been deleted.")  
 updateList()  
  
def create\_recipe():  
  
 # Creating a new window to create a new recipe  
 recipe\_create = Toplevel(recipeList)  
 recipe\_create.title('Create Recipe')  
 recipe\_create.geometry('1024x768')  
 recipe\_create.configure(bg='#318AE4')  
 recipe\_create.resizable(True, True)  
  
 # Label title  
 lblTitle = Label(recipe\_create, text='Create Recipe', font=('arial', 30, 'bold'), fg='black', bg='grey')  
 lblTitle.pack(pady=30)  
  
 # Label for name  
 lblName = Label(recipe\_create, text='Name: ', font=('arial', 15, 'bold'), fg='black', bg='grey')  
 lblName.pack(pady=15)  
  
 # Entry box for name  
 entry\_name = Entry(recipe\_create, width=15, bd=2, font=('arial', 30))  
 entry\_name.pack()  
  
 # Label for ingredients  
 lblIngredients = Label(recipe\_create, text='Ingredients: ', font=('arial', 15, 'bold'), fg='black', bg='grey')  
 lblIngredients.pack(pady=25)  
  
 # Text area for ingredients  
 text\_ingredients = Text(recipe\_create, width=25, height=10)  
 text\_ingredients.pack()  
  
 # Label for instructions  
 lblInstructions = Label(recipe\_create, text='Instructions: ', font=('arial', 15, 'bold'), fg='black', bg='grey')  
 lblInstructions.pack(pady=25)  
  
 # Text area for instructions  
 text\_instructions = Text(recipe\_create, width=100, height=10)  
 text\_instructions.pack()  
  
 # Creating a button to create the recipe  
 button\_create = Button(recipe\_create, text='Create Recipe', width=11, height=2, borderwidth=2,  
 command=lambda: create\_recipe\_in\_database(entry\_name.get(),  
 text\_ingredients.get("1.0", END),  
 text\_instructions.get("1.0", END)))  
 button\_create.pack()  
  
 def create\_recipe\_in\_database(name, ingredients, instructions):  
 # Connecting to the database  
 connection = sqlite3.connect('recipes.sqlite')  
 cursor = connection.cursor()  
  
 # Creating a new recipe  
 cursor.execute('INSERT INTO recipes (name, ingredients, instructions) VALUES (?, ?, ?)',  
 (name, ingredients, instructions))  
 connection.commit()  
  
 # Closing the connection  
 connection.close()  
 # Creating a success message  
 messagebox.showinfo('Success', 'Recipe created successfully!')  
 updateList()  
  
  
def updateList():  
 connection = sqlite3.connect('recipes.sqlite')  
 cursor = connection.cursor()  
  
 # Retrieving the recipes from the database  
 cursor.execute('SELECT \* FROM recipes')  
 recipes = cursor.fetchall()  
  
 # Closing the connection  
 connection.close()  
  
 listbox.delete(0, tk.END)  
  
 for recipe in recipes:  
 listbox.insert(END, recipe[0])  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main\_screen()