



RECIPE BOOK

S Devipriya

REG212406379

Certificate Course in Coding Skills

ASAP Kerala

CONTENTS

1. Introduction	3
1.1 Scope	3
1.2 Program objectives	3
2. System Requirements	4
2.1 Software Requirements	4
2.2 Python Libraries	4
3. Program Description	5
3.1 Functional Description	5
3.2 Structure Chart	8
3.3 Database Schema	9
4. Source Code	10
4.1 Source Code	10
4.2 Github Repository	44
5. Example Outputs	45
6. Conclusion	55
7. Future Scope	56
8. References	57

1. INTRODUCTION

1.1 Scope

This program is created for the final project in Certificate course in Coding Skills offered by ASAP Kerala. The scope of this project is to demonstrate an understanding of python fundamentals and basics of software design. This project report outlines the process of developing this project and its functionalities.

1.2 Program Objectives

This program called Recipe Book is a Community Recipe Management System. It allows users to input their recipes and adds it to a common database. This way other users can search for any recipes present in the system. Additionally, it offers functionality to edit your own recipes or rate others recipes. The program also maintains some data analytics regarding number of recipes contributed or rated by a user, as well as most popular recipes.

2. SYSTEM REQUIREMENTS

2.1 Software Requirements

Operating System: Windows 8+ or Linux

Database: MySQL Workbench 8.0 CE

Programming Software: Python 3.13.1 (3.1 or above)

2.2 Python Libraries

tkinter: messagebox, ttk, PhotoImage

mysql.connector

sqlalchemy

pandas

matplotlib.pyplot

matplotlib.figure: Figure

matplotlib.backends.backend_tkagg: FigureCanvasTkAgg

3. PROGRAM DESCRIPTION

3.1 Functional Description

The Recipe Book offers the following functionalities:

A) User Registration and Login

User should be able to register themselves with a username and password. Following this they should be able to login to the Recipe Book during which their credentials have to be validated against stored user authentication details.

B) Search for Recipes

Recipe Book being a community Recipe Management System must have foremost a searching for recipes functionality. Users can search for recipes in three ways:

- a) Search by name of a recipe.
- b) Search by tags assigned to a recipe such as sweet, dairy-free, meal, dessert, etc.
- c) Search without any input to receive a random recipe.

C) Manage Recipes

Users can manage their own recipes. This includes functionalities for viewing, adding, modifying or deleting their own recipes.

a) View recipes entered by user. This should show all details of recipes entered by the user in a table format.

b) Add a new recipe that the user wants to contribute to Recipe Book.

c) Modify already entered recipe's ingredients, instructions and tags. This allows for error-correction or other recipe adjustments.

d) Delete a particular recipe entered by the user. Users should be prompted with which recipe to delete.

D) Rate Recipes

To enhance community aspect, there must be functionality to rate any recipes available in the system. Users should also be able to see recipes they have rated.

a) Rate recipes by entering a particular recipe or a random recipe. There must be a way to provide a random recipe to rate

to ensure a smoother experience for users who don't want to look up which recipe to rate next.

b) User must be able to see a list of recipes they have rated.

E) Recipe Book Statistics

Recipe Book should also provide some stats to curious users.

The stats planned currently include:

a) Top 5 popular recipes.

b) Users who contributed most recipes.

c) Users who rated most recipes.

This information can be displayed graphically or in tabular form.

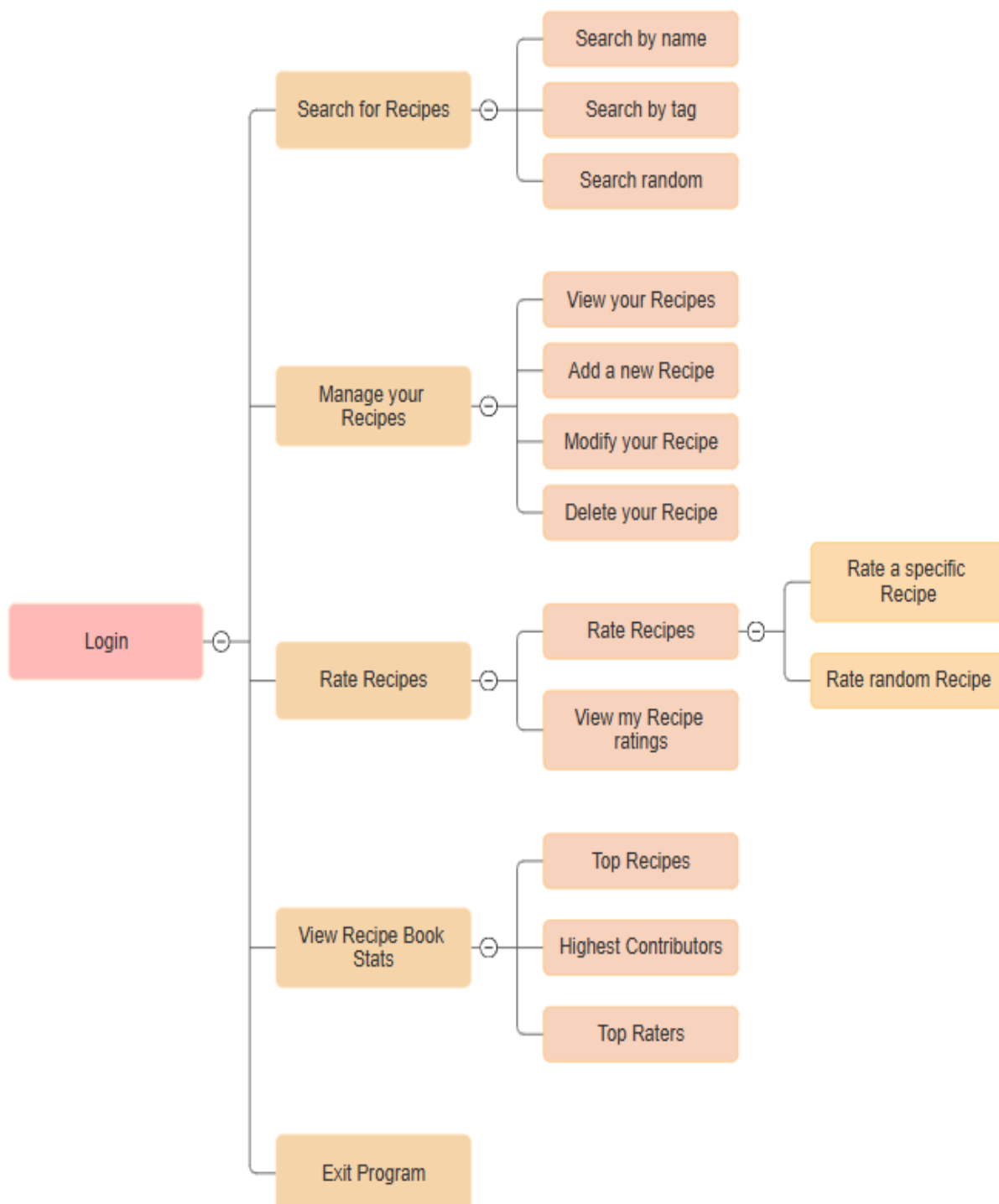
It can be useful to display user's own stats regardless of if they make top bracket or not for comparison purposes.

F) Exit and Back

The program should have basic navigation functions of going back to previous screen, wherever applicable and exiting the program altogether.

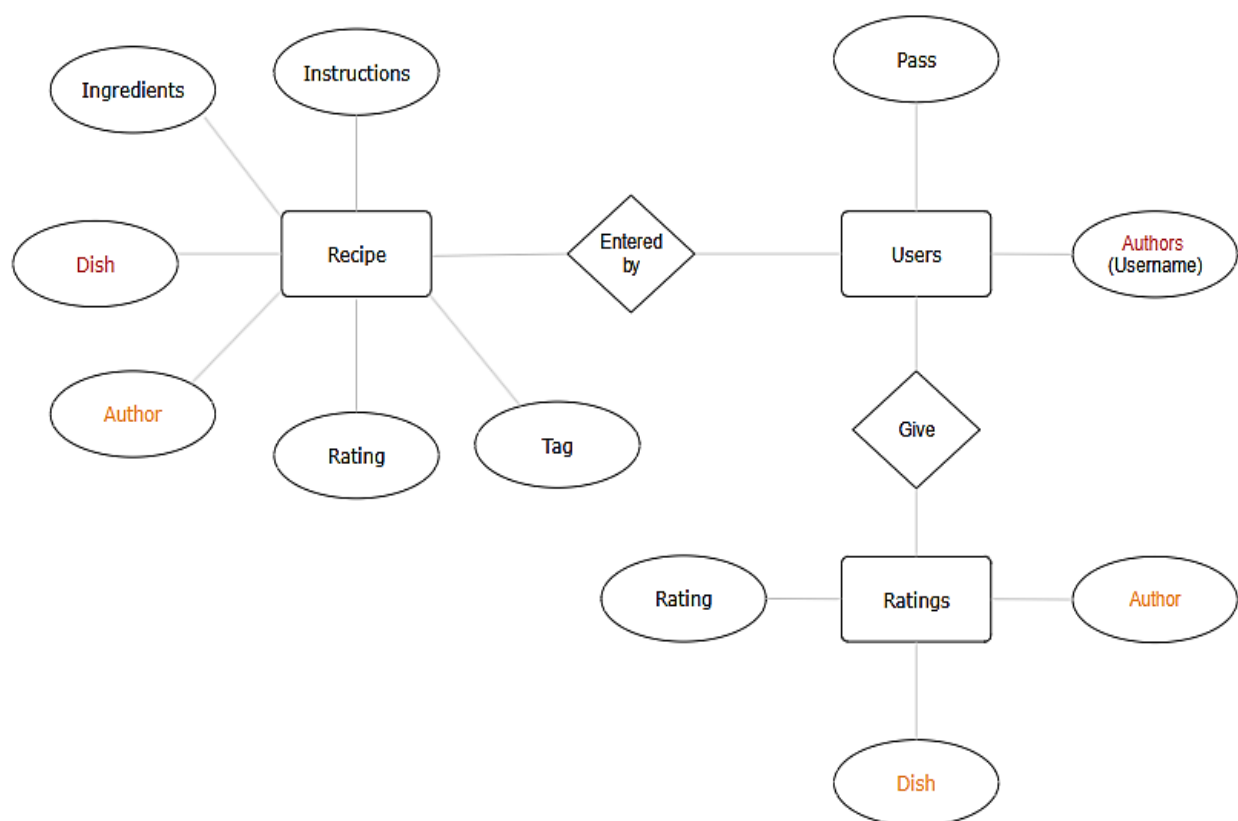
3.2 Structure Chart

Based on the identified functional requirements, a program design was created. The design can be summarized in the following structure chart:



3.3 Database Schema

Based on the requirements identified, there would require three different tables: one for storing recipe details, other for storing user authentication details and last one for storing recipes rated by users. A rough ER diagram is shown below:



Based on this database is designed. The database schema of the database “recipebook” used in this program is given below.

recipes

Dish	Author	Ingredients	Instructions	Tag	Rating
------	--------	-------------	--------------	-----	--------

Dish: Primary

users

Author	Pass
--------	------

Author: Primary

ratings

Author	Dish	Rating
--------	------	--------

Author + Dish: Primary

For ease of use the database and constituent tables are created via the program itself. Only connection to a database named “recipebook” is required to run the program.

4. SOURCE CODE

4.1 Source Code

Source code of the program is provided below.

```
import tkinter as tk
from tkinter import messagebox
from tkinter import ttk
from tkinter import PhotoImage
import mysql.connector as msc
import sqlalchemy as sq
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

def create_db():
    try:
        con = msc.connect(host='localhost',user='root',password='2023')
        cur = con.cursor()
        cur.execute("CREATE DATABASE IF NOT EXISTS recipebook;")
        con.commit()
        con.close()
        con = msc_connect_db()
        cur = con.cursor()
        cur.execute("CREATE TABLE IF NOT EXISTS users (Author VARCHAR(25),
Pass VARCHAR(25), PRIMARY KEY(Author));")
        con.commit()
        cur.execute("CREATE TABLE IF NOT EXISTS recipes (Dish VARCHAR(50), Tag
VARCHAR(100), Rating FLOAT, Ingredients VARCHAR(200), Instructions
VARCHAR(500), Author VARCHAR(25), PRIMARY KEY(Dish));")
        con.commit()
        cur.execute("CREATE TABLE IF NOT EXISTS ratings (Author
VARCHAR(25),Dish VARCHAR(25),Rating FLOAT, PRIMARY KEY(Author,Dish));")
        con.commit()
        con.close()
    except msc.Error as err:
        messagebox.showerror("Database couldnot be created",f"Error: {err}")

# Database connection
def connect_db():
```

```

        return
sq.create_engine("mysql+mysqlconnector://root:2023@localhost/recipebook")
def msc_connect_db():
    return
msc.connect(host='localhost',user='root',password='2023',database='recipebook'
)

# Registration function
def register():
    username = reg_username.get()
    password = reg_password.get()

    data = {'Author':[username],'Pass':[password]}
    if not username or not password:
        messagebox.showwarning("Input Error", "Both username and password are
required")
        return
    try:
        cur = connect_db()
        df = pd.DataFrame(data)
        df.to_sql('users',cur,if_exists='append',index=False)
        messagebox.showinfo("Success", "Registration Successful")
    except msc.Error as err:
        messagebox.showerror("Database Error",f"Error: {err}")
    reg_window.destroy()
    show_login_window()

# Login function
def login():
    username = login_username.get()
    password = login_password.get()

    cur = connect_db()
    q = f"SELECT * FROM users WHERE Author= '{username}' AND Pass=
'{password}';"
    result = pd.read_sql(q,cur)

    if not result.empty:
        if result.iloc[0,0]==username and result.iloc[0,1]==password:
            login_window.destroy()
            show_home_window(username)
        else:
            messagebox.showerror("Error", "Invalid Username or Password")

def show_window(title, size=(1024, 768)):
    window = tk.Tk()
    window.title(title)

```

```

window.minsize(*size)
return window

def show_frame( master, bg="#ffb5d1", relief=tk.FLAT, borderwidth=5):
    frame = tk.Frame(master=master, bg=bg, relief=relief,
borderwidth=borderwidth)
    frame.pack(fill=tk.BOTH, expand=True)
    return frame

def show_image(frame, image_path, relwidth=1, relheight=1, anchor="nw"):
    image = PhotoImage(file=image_path)
    image_lbl = tk.Label(frame, image=image)
    image_lbl.image = image # Keep a reference to avoid garbage collection
    image_lbl.place(x=0, y=0, relwidth=relwidth, relheight=relheight,
anchor=anchor)

def exit_app(win):
    win.destroy()
    thanks_window = show_window("Recipe Book")
    main_fr = show_frame(thanks_window)
    show_image(main_fr, "Recipebook.png")
    lbl_thanks= tk.Label(master= main_fr,text="Thank you for using Recipe
Book!",font=("Helvetica", 18),bg= "#ffb5d1",relief= tk.GROOVE,borderwidth= 5)
    lbl_thanks.pack(padx=50,pady=160)
    thanks_window.mainloop()

def show_welcome_window():
    global welcome_window
    try:
        reg_window.destroy()
    except:
        pass
    try:
        login_window.destroy()
    except:
        pass

    welcome_window = show_window("Recipe Book")
    main_fr = show_frame(welcome_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_in.pack(pady=200)
    fr_btn= tk.Frame(master= main_fr, bg= "#febebf")
    fr_btn.pack(pady=60)

```

```

    lbl_name= tk.Label(master= fr_in, text="Recipe Book!", font=("Helvetica",
20),bg= "#ffb5d1")
    btn_reg= tk.Button(master= fr_btn, text="Register", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=show_reg_window)
    btn_log= tk.Button(master= fr_btn, text="Login", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=show_login_window)

    lbl_name.grid(row=0,column=0)
    btn_reg.grid(row=0,column=2,padx=16)
    btn_log.grid(row=0,column=4,padx=5)

    welcome_window.mainloop()

def show_reg_window():
    global reg_window, reg_username, reg_password, ent_name, ent_pass
    try:
        welcome_window.destroy()
    except:
        pass

    reg_window = show_window("Recipe Book")
    main_fr = show_frame(reg_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_in.pack(pady=100)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=50)
    fr_btn= tk.Frame(master= main_fr, bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_reg= tk.Label(master= fr_in, text="Register", font=("Helvetica",
16),bg= "#ffb5d1")
    lbl_name= tk.Label(master= fr_box,text="Username:",bg= "#ffb5d1",
font=("Helvetica",14))
    reg_username = tk.StringVar()
    ent_name = tk.Entry(master= fr_box, textvariable=reg_username,
font=("Helvetica",14))
    lbl_pass= tk.Label(master= fr_box, text="Password:",bg= "#ffb5d1",
font=("Helvetica",14))
    reg_password = tk.StringVar()
    ent_pass = tk.Entry(master= fr_box, textvariable=reg_password, show='*',
font=("Helvetica",14))

    lbl_reg.grid(row=0,column=0)
    lbl_name.grid(row=5,column=2,pady=20,padx=5)

```

```

lbl_pass.grid(row=6,column=2,padx=5)
ent_name.grid(row=5,column=10,padx=5)
ent_pass.grid(row=6,column=10,pady=20,padx=5)

btn_reg= tk.Button(master= fr_btn, text="Register", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=register)
btn_log= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14),
command=show_welcome_window)

btn_reg.grid(row=0,column=2,padx=5)
btn_log.grid(row=0,column=4,padx=5)

reg_window.mainloop()

# Function to show the login window
def show_login_window():
    global login_window, login_username, login_password, ent_lname, ent_lpass,
    reg_window

    try:
        reg_window.destroy()
    except:
        pass
    try:
        welcome_window.destroy()
    except:
        pass

    login_window = show_window("Recipe Book")
    main_fr = show_frame(login_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_in.pack(pady=100)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=50)
    fr_btn= tk.Frame(master= main_fr, bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_log= tk.Label(master= fr_in, text="Login", font=("Helvetica", 16),bg=
"#ffb5d1")
    lbl_lname= tk.Label(master= fr_box,text="Username:",bg= "#ffb5d1",
font=("Helvetica",14))
    login_username = tk.StringVar()

```

```

    ent_lname = tk.Entry(master= fr_box, textvariable=login_username,
font=("Helvetica",14))
    lbl_lpass= tk.Label(master= fr_box, text="Password:",bg= "#ffb5d1",
font=("Helvetica",14))
    login_password = tk.StringVar()
    ent_lpass = tk.Entry(master= fr_box, textvariable=login_password,
show='*', font=("Helvetica",14))

    lbl_log.grid(row=0,column=0)
    lbl_lname.grid(row=5,column=2,pady=20,padx=5)
    lbl_lpass.grid(row=6,column=2,padx=5)
    ent_lname.grid(row=5,column=10,padx=5)
    ent_lpass.grid(row=6,column=10,pady=20,padx=5)

    btn_login= tk.Button(master= fr_btn, text="Login", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=login)
    btn_back= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14),
command=show_welcome_window)

    btn_login.grid(row=0,column=2,padx=5)
    btn_back.grid(row=0,column=4,padx=5)

    login_window.mainloop()

# Function to show the home window
def show_home_window(username):
    global home_window

    try:
        manage_window.destroy()
    except:
        pass
    try:
        search_window.destroy()
    except:
        pass
    try:
        rate_window.destroy()
    except:
        pass
    try:
        stats_window.destroy()
    except:
        pass

    home_window = show_window("Recipe Book")
    main_fr = show_frame(home_window)

```



```

show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_name= tk.Frame(master= main_fr,bg= "#febebf")
fr_name.pack(anchor= tk.W, padx=10)
fr_btns= tk.Frame(master= main_fr,bg=
"#febebf",borderwidth=5,relief=tk.SUNKEN)
fr_btns.pack(pady=100)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=30)

lbl_wel= tk.Label(master= fr_in, text="Welcome to Recipe Book!",
font=("Helvetica", 18),bg= "#ffb5d1")
lbl_name= tk.Label(master= fr_name,text=f"Good to see you again
{username}! What would you like to do today?", font=("Helvetica",14),bg=
"#ffb5d1",relief=tk.GROOVE,borderwidth=5,)
btn1= tk.Button(master= fr_btns, text="Search for a
recipe",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1", command= lambda:show_search_window(username))
btn2= tk.Button(master= fr_btns, text="Manage your
recipes",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1", command= lambda: show_manage_window(username))
btn3= tk.Button(master= fr_btns, text="Rate recipes",font=("Helvetica",
14), relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:show_rate_window(username))
btn4= tk.Button(master= fr_btns, text="View Recipe Book
stats",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1", command= lambda:show_stats_window(username))
btn5= tk.Button(master= fr_ext, text="Exit",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:exit_app(home_window))

lbl_wel.grid(row=0,column=0,padx=10,pady=10)
lbl_name.grid(row=0,column=0)
btn1.grid(row=2,column=0,padx=10,pady=10)
btn2.grid(row=3,column=0,padx=10,pady=10)
btn3.grid(row=4,column=0,padx=10,pady=10)
btn4.grid(row=5,column=0,padx=10,pady=10)
btn5.grid(row=0,column=0)

home_window.mainloop()

#Function to show the manage_widow
def show_manage_window(username):
    global home_window, manage_window

```

```

try:
    home_window.destroy()
except:
    pass
try:
    showres_window.destroy()
except:
    pass

manage_window = show_window("Recipe Book")
main_fr = show_frame(manage_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_btns= tk.Frame(master= main_fr,bg=
"#febebf",borderwidth=5,relief=tk.SUNKEN)
fr_btns.pack(pady=100)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=30)

lbl_ch= tk.Label(master= fr_in, text="Manage Your Recipes",
font=("Helvetica", 16),bg= "#ffb5d1")
btn1= tk.Button(master= fr_btns, text="View your
recipes",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1", command= lambda:show_recipe(username))
btn2= tk.Button(master= fr_btns, text="Add a new
recipe",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command=lambda:add_new_recipe(username))
btn3= tk.Button(master= fr_btns, text="Modify your
recipe",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command= lambda:mod_recipe(username))
btn4= tk.Button(master= fr_btns, text="Delete your
recipe",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command= lambda:del_recipe(username))
btn5= tk.Button(master= fr_ext, text="Exit",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:exit_app(manage_window))
btn6= tk.Button(master= fr_ext, text="Back",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:show_home_window(username))

lbl_ch.grid(row=0,column=0,padx=10,pady=10)
btn1.grid(row=2,column=0,padx=10,pady=10)
btn2.grid(row=3,column=0,padx=10,pady=10)
btn3.grid(row=4,column=0,padx=10,pady=10)
btn4.grid(row=5,column=0,padx=10,pady=10)

```

```

btn5.grid(row=0,column=2, padx=5)
btn6.grid(row=0,column=4, padx=5)

manage_window.mainloop()

def add_new_recipe(username):
    global manage_window

    try:
        manage_window.destroy()
    except:
        pass

    addres_window = show_window("Recipe Book")
    main_fr = show_frame(addres_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=50)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Enter New Recipe",
font=("Helvetica", 16),bg= "#ffb5d1")
    lbl_dish= tk.Label(master= fr_box,text="Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))
    lbl_tags= tk.Label(master= fr_box,text="Tags:",bg= "#ffb5d1",
font=("Helvetica",14))
    lbl_ing= tk.Label(master= fr_box,text="Ingredients:",bg= "#ffb5d1",
font=("Helvetica",14))
    lbl_instr= tk.Label(master= fr_box,text="Instructions:",bg= "#ffb5d1",
font=("Helvetica",14))
    global dish,tag,ing,instr
    dish = tk.StringVar()
    tag = tk.StringVar()
    ing = tk.StringVar()
    instr = tk.StringVar()

    ent_dish = tk.Entry(master= fr_box, textvariable=dish,width= 20,
font=("Helvetica",14))
    ent_tags = tk.Entry(master= fr_box, textvariable=tag,width= 20,
font=("Helvetica",14))
    txt_ing = tk.Text(master= fr_box,height= 5,width= 20,
font=("Helvetica",14))

```

```

txt_instr = tk.Text(master= fr_box,height= 5,width= 20,
font=("Helvetica",14))

lbl_ch.grid(row=0,column=0)
lbl_dish.grid(row=5,column=2,pady=20,padx=5)
lbl_tags.grid(row=6,column=2,pady= 20,padx=5)
lbl_ing.grid(row=8,column=2,pady= 20,padx=5)
lbl_instr.grid(row=13,column=2,pady= 20,padx=5)
ent_dish.grid(row=5,column=10,pady= 20,padx=5)
ent_tags.grid(row=6,column=10,pady= 20,padx=5)
txt_ing.grid(row=8,column=10,pady= 20,padx=5)
txt_instr.grid(row=13,column=10,padx=5,pady= 20)

def addres(username):
    dish= ent_dish.get()
    tag= ent_tags.get()
    ing= txt_ing.get("1.0",tk.END)
    instr= txt_instr.get("1.0",tk.END)
    cur = connect_db()
    data =
{'Dish':[dish],'Tag':[tag],'Rating':[0.0],'Ingredients':[ing],'Instructions':[
instr],'Author':[username]}
    df = pd.DataFrame(data)
    df.to_sql('recipes',cur,if_exists='append',index=False)
    messagebox.showinfo("Success","Recipe added successfully!")
    addres_window.destroy()
    show_home_window(username)

    btn_reg= tk.Button(master= fr_btn, text="Add", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
addres(username))

    btn_reg.grid(row=0,column=2)

    addres_window.mainloop()

def show_recipe(username):

    global showres_window
    try:
        manage_window.destroy()
    except:
        pass

    showres_window = show_window("Recipe Book")
    main_fr = show_frame(showres_window)
    show_image(main_fr, "Recipebook.png")

```

```

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=50)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=40)

lbl_ch= tk.Label(master= fr_in, text="Your Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_ch.grid(row=0,column=0,padx=10,pady=10)

tree = ttk.Treeview(master= fr_box, style='Treeview.Heading')
style = ttk.Style()
style.configure("Treeview.Heading",bg= "#ffb5d1")
tree.pack(pady=10)

def showres(tree):
    cur = connect_db()
    q = f"SELECT * from recipes WHERE Author='{username}';"
    df = pd.read_sql_query(q,cur)
    tree["columns"] = list(df.columns)
    tree["show"] = "headings"
    for col in df.columns:
        tree.heading(col, text=col)
        tree.column(col, width=160)
    for index, row in df.iterrows():
        tree.insert("", tk.END, values=list(row))
    print(df.iterrows())

fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

btn_reg= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_manage_window(username))
btn_reg.grid(row=0,column=2)

showres(tree)
showres_window.mainloop()

def del_recipe(username):
    try:
        manage_window.destroy()
    except:
        pass

def delres(username):
    cur = connect_db()

```

```

q = f"SELECT * FROM recipes WHERE Author='{username}';"
df = pd.read_sql_query(q,cur)
delf = df[df['Dish']==dish].index
df.drop(delf, inplace=True)
df.to_sql('recipes',cur,if_exists='replace',index=False)
messagebox.showinfo("Success","Recipe deleted successfully!")
delres_window.destroy()
show_manage_window()

delres_window = show_window("Recipe Book")
main_fr = show_frame(delres_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_del= tk.Frame(master= main_fr,relief=tk.GROOVE,bg= "#ffb5d1")
fr_del.pack(fill= tk.X)
fr_box= tk.Frame(master= main_fr,bg= "#ffb5d1",borderwidth=5,relief=
tk.GROOVE)
fr_box.pack(pady=50)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Recipe Deletion", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_ch.grid(row=0,column=0,padx=10,pady=10)
lbl_del= tk.Label(master= fr_del,text= "Enter name of recipe to
delete",font=("Helvetica",14),bg= "#ffb5d1",relief=tk.GROOVE, borderwidth=5)
lbl_dish= tk.Label(master= fr_box,text= "Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))

dish = tk.StringVar()
ent_dish= tk.Entry(master= fr_box,textvariable= dish, width=20,
font=("Helvetica",14))
dish = dish.get()

btn_del= tk.Button(master= fr_btn, text="Delete", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
delres(username))

lbl_del.grid(row=0,column=0)
lbl_dish.grid(row=10,column=2,pady= 20,padx=5)
ent_dish.grid(row=10,column=10,pady= 20,padx=5)
btn_del.grid(row= 0, column= 2)

delres_window.mainloop()

```

```

def mod_recipe(username):

    try:
        manage_window.destroy()
    except:
        pass

    modres_window = show_window("Recipe Book")
    main_fr = show_frame(modres_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=50)
    fr_mod= tk.Frame(master= main_fr,relief=tk.GROOVE,bg= "#ffb5d1")
    fr_mod.pack(fill= tk.X)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=40)
    fr_tab= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_tab.pack(pady=40)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Modify Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
    lbl_ch.grid(row=0,column=0,padx=10,pady=10)
    lbl_mod= tk.Label(master= fr_mod,text= "Enter name of recipe to
modify",font=("Helvetica",14),bg= "#ffb5d1",relief=tk.GROOVE, borderwidth=5)
    lbl_dish= tk.Label(master= fr_box,text= "Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))
    global dish
    dish = tk.StringVar()
    ent_dish= tk.Entry(master= fr_box,textvariable= dish, width=20,
font=("Helvetica",14))

    btn_view= tk.Button(master= fr_btn, text="View", relief= tk.RAISED,
borderwidth=5, font=("Helvetica",14),bg= "#ffb5d1",command= lambda:
viewres(username))
    lbl_mod.grid()
    lbl_ch.grid(row=0,column=0,padx=5)
    lbl_dish.grid(row=10,column=2,pady= 20,padx=5)
    ent_dish.grid(row=10,column=10,pady= 20,padx=5)
    btn_view.grid(row= 0, column= 2, padx=5)

    def viewres(username):
        dish = ent_dish.get()

```

```

tree = ttk.Treeview(master= fr_tab,style='Treeview.Heading')
style = ttk.Style()
style.configure("Treeview.Heading",bg= "#ffb5d1")
tree.pack(pady=10)
cur = connect_db()
q = f"SELECT * FROM recipes WHERE Author=('{username}')) AND
Dish=('{dish}')";"
df = pd.read_sql_query(q,cur)
tree["columns"] = list(df.columns)
tree["show"] = "headings"
for col in df.columns:
    tree.heading(col, text=col)
    tree.column(col, width=160)
for index, row in df.iterrows():
    tree.insert("", tk.END, values=list(row))
print(df.iterrows())

btn_mod= tk.Button(master= fr_btn, text="Modify", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_modify_window(username,dish))
btn_mod.grid(row= 0, column= 4, padx=5)

def show_modify_window(username,dish):
    smodres_window = show_window("Recipe Book")
    main_fr = show_frame(smodres_window)

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=50)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text=f"Enter Modified Recipe for
{dish}", font=("Helvetica", 16),bg= "#ffb5d1")
    lbl_ing= tk.Label(master= fr_box,text="Ingredients:",bg= "#ffb5d1",
font=("Helvetica",14))
    lbl_instr= tk.Label(master= fr_box,text="Instructions:",bg= "#ffb5d1",
font=("Helvetica",14))
    global ing,instr
    ing = tk.StringVar()
    instr = tk.StringVar()

    txt_ing = tk.Text(master= fr_box,height= 5,width= 20,
font=("Helvetica",14))

```



```

        txt_instr = tk.Text(master= fr_box,height= 5,width= 20,
font=("Helvetica",14))

        lbl_ch.grid(row=0,column=0)
        lbl_ing.grid(row=0,column=2,pady= 20,padx=5)
        lbl_instr.grid(row=5,column=2,pady= 20,padx=5)
        txt_ing.grid(row=0,column=10,pady= 20,padx=5)
        txt_instr.grid(row=5,column=10,padx=5,pady= 20)

        def modres(username,dish):
            ing= txt_ing.get("1.0",tk.END)
            instr= txt_instr.get("1.0",tk.END)
            con = msc_connect_db()
            cur = con.cursor()
            q = f"UPDATE recipes SET
Ingredients=('{ing}'),Instructions=('{instr}') WHERE Author=('{username}') AND
Dish=('{dish}')";
            cur.execute(q)
            con.commit()
            con.close()
            messagebox.showinfo("Success","Recipe modified successfully!")
            smodres_window.destroy()
            modres_window.destroy()

            show_home_window(username)

        btn_reg= tk.Button(master= fr_btn, text="Modify", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
modres(username,dish))
        btn_reg.grid(row=0,column=2)

        smodres_window.mainloop()
        modres_window.mainloop()

def show_search_window(username):
    global home_window, search_window, search_by_name_window,
searchres_tag_window, show_surprise_window

    try:
        home_window.destroy()
    except:
        pass
    try:
        search_by_name_window.destroy()
    except:
        pass
    try:
        searchres_tag_window.destroy()

```

```

except:
    pass
try:
    show_surprise_window.destroy()
except:
    pass

search_window = show_window("Recipe Book")
main_fr = show_frame(search_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_btns= tk.Frame(master= main_fr,bg=
"#febebf",borderwidth=5,relief=tk.SUNKEN)
fr_btns.pack(pady=100)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=30)

lbl_ch= tk.Label(master= fr_in, text="Search Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
btn1= tk.Button(master= fr_btns, text="Search by name",font=("Helvetica",
14), relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:search_recipe_name(username))
btn2= tk.Button(master= fr_btns, text="Search by tag",font=("Helvetica",
14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command=lambda:search_recipe_tag(username))
btn3= tk.Button(master= fr_btns, text="Surprise me!",font=("Helvetica",
14), relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1",command =
lambda:surprise_recipe(username))
btn4= tk.Button(master= fr_ext, text="Exit",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:exit_app(search_window))
btn5= tk.Button(master= fr_ext, text="Back",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:show_home_window(username))

lbl_ch.grid(row=0,column=0,padx=10,pady=10)
btn1.grid(row=2,column=0,padx=10,pady=10)
btn2.grid(row=3,column=0,padx=10,pady=10)
btn3.grid(row=4,column=0,padx=10,pady=10)
btn4.grid(row=0,column=2,padx=5)
btn5.grid(row=0,column=4, padx=5)

search_window.mainloop()

def search_recipe_name(username):

```

```

try:
    search_window.destroy()
except:
    pass

searchres_name_window = show_window("Recipe Book")
main_fr = show_frame(searchres_name_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=30)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Recipe Search", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_dish= tk.Label(master= fr_box,text="Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))
global dish
dish = tk.StringVar()
ent_dish = tk.Entry(master= fr_box, textvariable=dish,width= 20,
font=("Helvetica",14))
btn_search = tk.Button(master= fr_btn, text="Search", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
search_by_name(dish))
lbl_ch.grid(row=0,column=0)
lbl_dish.grid(row=5,column=2,pady=20,padx=5)
ent_dish.grid(row=5,column=10,pady= 20,padx=5)
btn_search.grid(row=0,column=2)

def search_by_name(dish):
    global search_by_name_window
    cur = connect_db()
    dish = ent_dish.get()
    q = f"SELECT * FROM recipes WHERE Dish='{dish}';"
    df = pd.read_sql_query(q,cur)
    tag = df.iloc[0,1]
    rat = df.iloc[0,2]
    ing = df.iloc[0,3]
    instr = df.iloc[0,4]
    author = df.iloc[0,5]

    try:
        searchres_name_window.destroy()

```

```

except:
    pass

search_by_name_window = show_window("Recipe Book")
main_fr = show_frame(search_by_name_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=30)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Recipe Found!",
font=("Helvetica", 16),bg= "#ffb5d1")
txt_res = tk.Text(master= fr_box, bg= "#ffb5d1",
font=("Helvetica",14), width= 75, height= 20, padx= 10, pady= 10)
btn_back = tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_search_window(username))
scrollbar = ttk.Scrollbar(master= fr_box, orient='vertical',
command=txt_res.yview)
scrollbar.grid(row=0, column=1, sticky=tk.NS)
txt_res['yscrollcommand'] = scrollbar.set

txt_res.insert(tk.END,"Dish Name: ")
txt_res.insert(tk.END,dish)
txt_res.insert(tk.END,"\n\nTags: ")
txt_res.insert(tk.END,tag)
txt_res.insert(tk.END,"\n\nRating: ")
txt_res.insert(tk.END,rate)
txt_res.insert(tk.END,"\n\nIngredients\n\n")
txt_res.insert(tk.END,ing)
txt_res.insert(tk.END,"\n\nInstructions\n\n")
txt_res.insert(tk.END,instr)
txt_res.insert(tk.END,"\n\nAuthor: ")
txt_res.insert(tk.END,author)

lbl_ch.grid(row=0,column=0)
txt_res.grid(row=0,column=0,padx=5,pady=5)
btn_back.grid(row=0,column=0)

search_by_name_window.mainloop()

searchres_name_window.mainloop()

```

```

def search_recipe_tag(username):
    global searchres_tag_window
    try:
        search_window.destroy()
    except:
        pass

    searchres_tag_window = show_window("Recipe Book")
    main_fr = show_frame(searchres_tag_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=30)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)
    fr_tab= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_tab.pack(pady=40)
    fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
    fr_ext.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Recipe Search", font=("Helvetica",
16),bg= "#ffb5d1")
    lbl_ch.grid(row=0,column=0)
    lbl_tag= tk.Label(master= fr_box,text="Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))
    global tag
    tag = tk.StringVar()
    ent_tag = tk.Entry(master= fr_box, textvariable=tag,width= 20,
font=("Helvetica",14))
    btn_search = tk.Button(master= fr_btn, text="Search", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
search_by_tag(tag))
    lbl_ch.grid(row=0,column=0)
    lbl_tag.grid(row=5,column=2,pady=20,padx=5)
    ent_tag.grid(row=5,column=10,pady= 20,padx=5)
    btn_search.grid(row=0,column=2)

    def search_by_tag(tag):
        cur = connect_db()
        tag = ent_tag.get()
        q = f"SELECT * FROM recipes WHERE Tag LIKE '%{tag}%';"
        df = pd.read_sql_query(q,cur)

```

```

        if df.empty:
            messagebox.showinfo("Error","No recipes found with the given tag")
            searchres_tag_window.destroy()
            show_search_window(username)

        tree = ttk.Treeview(master= fr_tab,style='Treeview.Heading')
        style = ttk.Style()
        style.configure("Treeview.Heading",bg= "#ffb5d1")
        tree.pack(pady=10)
        tree["columns"] = list(df.columns)
        tree["show"] = "headings"
        for col in df.columns:
            tree.heading(col, text=col)
            tree.column(col, width=160)
        for index, row in df.iterrows():
            tree.insert("", tk.END, values=list(row))
        print(df.iterrows())

        btn_mod= tk.Button(master= fr_ext, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_search_window(username))
        btn_mod.grid(row= 0, column= 5)

        searchres_tag_window.mainloop()

def surprise_recipe(username):
    try:
        search_window.destroy()
    except:
        pass

    surprise_window = show_window("Recipe Book")
    main_fr = show_frame(surprise_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=30)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Surprise Recipe!",
font=("Helvetica", 16),bg= "#ffb5d1")
    style = ttk.Style()

```

```

        style.configure("TCheckbutton",background= "#ffb5d1",
font=("Helvetica",14))
    global chk
    chk = tk.IntVar()
    chk_ready= ttk.Checkbutton(master= fr_box,text="I'm ready to be
surprised!",style= 'TCheckbutton',variable= chk,onvalue=1,offvalue=0,command=
lambda: surpriseres(chk))
    lbl_ch.grid(row=0,column=0)
    chk_ready.grid(row=0,column=0,pady=10)

def surpriseres(chk):
    global show_surprise_window
    cur = connect_db()
    chk = chk.get()
    if chk == 0:
        messagebox.showinfo("Error","Please check the box to proceed")
        surprise_window.destroy()
        show_search_window(username)
    q = f"SELECT * FROM recipes ORDER BY RAND() LIMIT 1;"
    df = pd.read_sql_query(q,cur)
    dish = df.iloc[0,0]
    tag = df.iloc[0,1]
    rat = df.iloc[0,2]
    ing = df.iloc[0,3]
    instr = df.iloc[0,4]
    author = df.iloc[0,5]

    try:
        surprise_window.destroy()
    except:
        pass

    show_surprise_window = show_window("Recipe Book")
    main_fr = show_frame(show_surprise_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_box.pack(pady=30)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Surprise Recipe!",
font=("Helvetica", 16),bg= "#ffb5d1")

```

```

        txt_res = tk.Text(master= fr_box, bg= "#ffb5d1",
font=("Helvetica",14), width= 75, height= 20, padx= 10, pady= 10)
        btn_back = tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_search_window(username))
        scrollbar = ttk.Scrollbar(master= fr_box, orient='vertical',
command=txt_res.yview)
        scrollbar.grid(row=0, column=1, sticky=tk.NS)
        txt_res['yscrollcommand'] = scrollbar.set

        txt_res.insert(tk.END,"Dish Name: ")
        txt_res.insert(tk.END,dish)
        txt_res.insert(tk.END,"\n\nTags: ")
        txt_res.insert(tk.END,tag)
        txt_res.insert(tk.END,"\n\nRating: ")
        txt_res.insert(tk.END,rate)
        txt_res.insert(tk.END,"\n\nIngredients\n\n")
        txt_res.insert(tk.END,ing)
        txt_res.insert(tk.END,"\n\nInstructions\n\n")
        txt_res.insert(tk.END,instr)
        txt_res.insert(tk.END,"\n\nAuthor: ")
        txt_res.insert(tk.END,author)

        lbl_ch.grid(row=0,column=0)
        txt_res.grid(row=0,column=0,padx=5,pady=5)
        btn_back.grid(row=0,column=0)

        show_surprise_window.mainloop()

surprise_window.mainloop()

def show_rate_window(username):
    global home_window, rate_window

    try:
        home_window.destroy()
    except:
        pass
    try:
        rate_recipe_window.destroy()
    except:
        pass
    try:
        myrating_window.destroy()
    except:
        pass

    rate_window = show_window("Recipe Book")

```



```

main_fr = show_frame(rate_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_btns= tk.Frame(master= main_fr,bg=
"#febebf",borderwidth=5,relief=tk.SUNKEN)
fr_btns.pack(pady=100)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=30)

lbl_ch= tk.Label(master= fr_in, text="Recipe Ratings", font=("Helvetica",
16),bg= "#ffb5d1")
btn1= tk.Button(master= fr_btns, text="Rate Recipe",font=("Helvetica",
14), relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:rate_recipe(username))
btn2= tk.Button(master= fr_btns, text="My Ratings",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command=lambda:my_ratings(username))
btn3= tk.Button(master= fr_ext, text="Exit",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:exit_app(rate_window))
btn4= tk.Button(master= fr_ext, text="Back",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:show_home_window(username))

lbl_ch.grid(row=0,column=0,padx=10,pady=10)
btn1.grid(row=2,column=0,padx=10,pady=10)
btn2.grid(row=3,column=0,padx=10,pady=10)
btn3.grid(row=4,column=2,padx=5)
btn4.grid(row=4,column=4,padx=5)

rate_window.mainloop()

def rate_specific(username):
    global rate_specific_window
    try:
        rate_recipe_window.destroy()
    except:
        pass

    rate_specific_window = show_window("Recipe Book")
    main_fr = show_frame(rate_specific_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")

```

```

fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=10)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Rate Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_dish= tk.Label(master= fr_box,text="Dish name:",bg= "#ffb5d1",
font=("Helvetica",14))
dish = tk.StringVar()
ent_dish = tk.Entry(master= fr_box, textvariable= dish, width= 20,
font=("Helvetica",14))
btn_back = tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_rate_window(username))
btn_view = tk.Button(master= fr_btn, text="View", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
find_dish(dish))
lbl_ch.grid(row=0,column=0)
lbl_dish.grid(row=0,column=0,padx=5,pady=5)
ent_dish.grid(row=0,column=10,padx=5,pady=5)
btn_back.grid(row=0,column=2)
btn_view.grid(row=0,column=4, padx=25)

def find_dish(dish):
    cur = connect_db()
    dish = ent_dish.get()
    q = f"SELECT * FROM recipes WHERE Dish='{dish}';"
    df = pd.read_sql_query(q,cur)
    if df.empty:
        messagebox.showinfo("Error","No recipes found with the given
name")
        rate_specific_window.destroy()
        show_rate_window(username)
    else:
        display_and_rate(username,df)

rate_specific_window.mainloop()

def rate_random(username):
    global rate_random_window
    try:
        rate_recipe_window.destroy()
    except:
        pass

```

```

rate_random_window = show_window("Recipe Book")
main_fr = show_frame(rate_random_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=10)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Rate Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
btn_back = tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_rate_window(username))
btn_view = tk.Button(master= fr_btn, text="See a recipe", relief=
tk.RAISED, borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=
lambda: find_dish())
lbl_ch.grid(row=0,column=0)
btn_back.grid(row=0,column=2)
btn_view.grid(row=0,column=4, padx=25)

def find_dish():
    cur = connect_db()
    q = f"SELECT * FROM recipes ORDER BY RAND() LIMIT 1;"
    df = pd.read_sql_query(q,cur)
    if df.empty:
        messagebox.showinfo("Error","No recipes found")
        rate_specific_window.destroy()
        show_rate_window(username)
    else:
        display_and_rate(username,df)

rate_random_window.mainloop()

def display_and_rate(username,df):
    dish = df.iloc[0,0]
    tag = df.iloc[0,1]
    current_rating = float(df.iloc[0,2])
    ing = df.iloc[0,3]
    instr = df.iloc[0,4]
    author = df.iloc[0,5]

    try:
        rate_specific_window.destroy()

```

```

except:
    pass
try:
    rate_random_window.destroy()
except:
    pass

def rate(username,dish,new_rating,current_rating):
    con = msc_connect_db()
    cur = con.cursor()
    new_rating = float(ent_rating.get())
    if new_rating < 0 or new_rating > 5:
        messagebox.showerror("Error","Rating should be between 0.0 and
5.0")
        return
    if current_rating == 0:
        final_rating = new_rating
    else:
        final_rating = (current_rating+new_rating)/2
    q = f"UPDATE recipes SET Rating=('{final_rating}') WHERE
Dish=('{dish}');"
    cur.execute(q)
    con.commit()
    q = f"INSERT INTO ratings (Author,Dish,Rating)
VALUES('{username}','{dish}','{new_rating}');"
    cur.execute(q)
    con.commit()
    con.close()

    messagebox.showinfo("Success","Recipe rated successfully!")
    viewnrate_window.destroy()
    show_rate_window(username)

viewnrate_window = show_window("Recipe Book")
main_fr = show_frame(viewnrate_window)
show_image(main_fr, "Recipebook.png")

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=10)
fr_btn= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_btn.pack(pady=10)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=20)

```

```

    lbl_ch= tk.Label(master= fr_in, text="Rate this Recipe",
font=("Helvetica", 16),bg= "#ffb5d1")
    txt_res = tk.Text(master= fr_box, bg= "#ffb5d1", font=("Helvetica",14),
width= 75, height= 10, padx=10, pady= 10)
    scrollbar = ttk.Scrollbar(master= fr_box, orient='vertical',
command=txt_res.yview)
    scrollbar.grid(row=0, column=1, sticky=tk.NS, rowspan=10)
    txt_res['yscrollcommand'] = scrollbar.set
    lbl_rating = tk.Label(master= fr_btn,text="Rating (0.0 to 5.0):",bg=
"#ffb5d1", font=("Helvetica",14))
    rating = tk.StringVar()
    ent_rating = tk.Entry(master= fr_btn,width= 20,textvariable= rating,
font=("Helvetica",14))
    btn_rate= tk.Button(master= fr_ext, text="Rate", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
rate(username,dish,rating,current_rating))

    lbl_ch.grid(row=0,column=0)
    txt_res.grid(row=2,column=0,padx=5,pady=5)
    lbl_rating.grid(row=0,column=0,padx=5,pady=5)
    ent_rating.grid(row=0,column=10,padx=5,pady=5)
    btn_rate.grid(row=0,column=0)

    txt_res.insert(tk.END,"Dish Name: ")
    txt_res.insert(tk.END,dish)
    txt_res.insert(tk.END,"\n\nTags: ")
    txt_res.insert(tk.END,tag)
    txt_res.insert(tk.END,"\n\nIngredients\n\n")
    txt_res.insert(tk.END,ing)
    txt_res.insert(tk.END,"\n\nInstructions\n\n")
    txt_res.insert(tk.END,instr)
    txt_res.insert(tk.END,"\n\nAuthor: ")
    txt_res.insert(tk.END,author)

    viewnrate_window.mainloop()

def rate_recipe(username):
    global rate_recipe_window

    try:
        rate_window.destroy()
    except:
        pass

    rate_recipe_window = show_window("Recipe Book")
    main_fr = show_frame(rate_recipe_window)
    show_image(main_fr, "Recipebook.png")

```

```

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_btn= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_btn.pack(pady=100)
fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
fr_ext.pack(pady=30)

lbl_ch= tk.Label(master= fr_in, text="Rate Recipes", font=("Helvetica",
16),bg= "#ffb5d1")
btn_one= tk.Button(master= fr_btn, text="Rate a specific recipe", relief=
tk.RAISED, borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=
lambda: rate_specific(username))
btn_random= tk.Button(master= fr_btn, text="Rate a random recipe", relief=
tk.RAISED, borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=
lambda: rate_random(username))
btn_exit= tk.Button(master= fr_ext, text="Exit", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
exit_app(rate_recipe_window))
btn_back= tk.Button(master= fr_ext, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_rate_window(username))

lbl_ch.grid(row=0, column=0, padx= 10, pady= 10)
btn_one.grid(row=0, column=0, padx= 10, pady= 10)
btn_random.grid(row=1, column=0, padx= 10, pady= 10)
btn_exit.grid(row=0, column=2, padx= 5)
btn_back.grid(row=0, column=4, padx=5)

rate_recipe_window.mainloop()

def my_ratings(username):
    global myrating_window
    try:
        rate_window.destroy()
    except:
        pass

    myrating_window = show_window("Recipe Book")
    main_fr = show_frame(myrating_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=50)

```

```

fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_box.pack(pady=40)
fr_tab= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
fr_tab.pack(pady=40)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="My Ratings", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_ch.grid(row=0,column=0,padx=10,pady=10)
btn_view= tk.Button(master= fr_box, text="View my ratings", relief=
tk.RAISED, borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command=
lambda: viewmyratings(username))

lbl_ch.grid(row=0,column=0,padx=5)
btn_view.grid(row=0,column=0)

def viewmyratings(username):
    tree = ttk.Treeview(master= fr_tab,style='Treeview.Heading')
    style = ttk.Style()
    style.configure("Treeview.Heading",bg= "#ffb5d1")
    tree.pack(pady=10)
    cur = connect_db()
    q = f"SELECT * FROM ratings WHERE Author=('{username}')";
    df = pd.read_sql_query(q,cur)
    tree["columns"] = list(df.columns)
    tree["show"] = "headings"
    for col in df.columns:
        tree.heading(col, text=col)
        tree.column(col, width=160)
    for index, row in df.iterrows():
        tree.insert("", tk.END, values=list(row))
    print(df.iterrows())

    btn_back= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_rate_window(username))
    btn_back.grid(row= 0,column=0)

myrating_window.mainloop()

def show_stats_window(username):
    global stats_window
    try:
        home_window.destroy()
    except:

```

```

        pass
    try:
        recipes_stat_window.destroy()
    except:
        pass
    try:
        authors_stat_window.destroy()
    except:
        pass
    try:
        raters_stat_window.destroy()
    except:
        pass

    stats_window = show_window("Recipe Book")
    main_fr = show_frame(stats_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_btns= tk.Frame(master= main_fr,bg=
"#febebf",borderwidth=5,relief=tk.SUNKEN)
    fr_btns.pack(pady=100)
    fr_ext= tk.Frame(master= main_fr,bg= "#febebf")
    fr_ext.pack(pady=30)

    lbl_ch= tk.Label(master= fr_in, text="Statistics", font=("Helvetica",
16),bg= "#ffb5d1")
    btn1= tk.Button(master= fr_btns, text="Top Rated
Recipes",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1", command= lambda: recipes_stat(username))
    btn2= tk.Button(master= fr_btns, text="Most Contributed
Authors",font=("Helvetica", 14), relief= tk.RAISED, borderwidth=5, bg=
"#ffb5d1",command=lambda: authors_stat(username))
    btn3= tk.Button(master= fr_btns, text="Top Raters",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1",command=lambda:
raters_stats(username))
    btn4= tk.Button(master= fr_ext, text="Exit",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command=
lambda:exit_app(stats_window))
    btn5= tk.Button(master= fr_ext, text="Back",font=("Helvetica", 14),
relief= tk.RAISED, borderwidth=5, bg= "#ffb5d1", command= lambda:
show_home_window(username))

    lbl_ch.grid(row=0,column=0,padx=10,pady=10)
    btn1.grid(row=0,column=0,padx=10,pady=10)
    btn2.grid(row=1,column=0,padx=10,pady=10)

```



```

btn3.grid(row=2,column=0,padx=10,pady=10)
btn4.grid(row=0,column=2,padx=5)
btn5.grid(row=0,column=4,padx=5)

stats_window.mainloop()

def recipes_stat(username):
    global recipes_stat_window
    try:
        stats_window.destroy()
    except:
        pass

    def get_topRated():
        cur = connect_db()
        q = f"SELECT * FROM recipes ORDER BY Rating DESC LIMIT 5;"
        df = pd.read_sql_query(q,cur)
        return df

    recipes_stat_window = show_window("Recipe Book")
    main_fr = show_frame(recipes_stat_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5,width= 60, height= 50)
    fr_box.pack(pady=10)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

    lbl_ch= tk.Label(master= fr_in, text="Top Rated Recipes",
font=("Helvetica", 16),bg= "#ffb5d1")
    lbl_ch.grid(row=0,column=0)

    df = get_topRated()
    fig= plt.figure(figsize=(6,5))
    ax= fig.add_subplot(111)
    ax.bar(df["Dish"],df["Rating"],color= "pink",edgecolor= "black",width=0.5)
    ax.set_xlabel("Dish", fontdict= {"fontsize": 10, "fontweight": "bold"})
    ax.set_ylabel("Rating", fontdict= {"fontsize": 10, "fontweight": "bold"})
    ax.set_title("Top Rated Recipes", fontsize= 12, fontweight= "bold", color=
"lightcoral")

    canvas= FigureCanvasTkAgg(fig, master= fr_box)
    canvas_widget= canvas.get_tk_widget()

```

```

        canvas_widget.pack(side=tk.TOP,fill=tk.BOTH,expand=True, padx= 10, pady=
10)
        canvas.draw()

        btn_back= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_stats_window(username))
        btn_back.grid(row= 0,column=0)

        recipes_stat_window.mainloop()

def authors_stat(username):
    global authors_stat_window
    try:
        stats_window.destroy()
    except:
        pass

    def get_most_contributed():
        cur = connect_db()
        q = f"SELECT Author,COUNT(Dish) as Contributions FROM recipes GROUP BY
Author ORDER BY Contributions DESC LIMIT 5;"
        df = pd.read_sql_query(q,cur)
        return df

    def get_my_count(username):
        cur = connect_db()
        q = f"SELECT COUNT(Dish) as Contributions FROM recipes WHERE
Author=('{username}')";
        df = pd.read_sql_query(q,cur)
        return df.iloc[0,0]

    authors_stat_window = show_window("Recipe Book")
    main_fr = show_frame(authors_stat_window)
    show_image(main_fr, "Recipebook.png")

    fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
    fr_in.pack(pady=70)
    fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5, width= 60, height= 50)
    fr_box.pack(pady=10)
    fr_tab= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5)
    fr_tab.pack(pady=40)
    fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
    fr_btn.pack(pady=10)

```

```

    lbl_ch= tk.Label(master= fr_in, text="Our Best Contributors",
font=("Helvetica", 16),bg= "#ffb5d1")
    lbl_ch.grid(row=0,column=0)
    df = get_most_contributed()
    tree = ttk.Treeview(master= fr_box,style='Treeview.Heading')
    style = ttk.Style()
    style.configure("Treeview.Heading",bg= "#ffb5d1")
    tree.pack(pady=10, fill= tk.BOTH, expand= True)
    tree["columns"] = list(df.columns)
    tree["show"] = "headings"
    for col in df.columns:
        tree.heading(col, text=col)
        tree.column(col, width=160)
    for index, row in df.iterrows():
        tree.insert("", tk.END, values=list(row))
    print(df.iterrows())

    my_count = get_my_count(username)
    lbl_my_count = tk.Label(master= fr_tab, text=f"And ofcourse, you too! Your
contributions: {my_count}", font=("Helvetica", 14), bg= "#ffb5d1")
    lbl_my_count.grid(row=0,column=0,padx=10,pady=10)

    btn_back= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_stats_window(username))
    btn_back.grid(row= 0,column=0)

    authors_stat_window.mainloop()

def raters_stats(username):
    global raters_stat_window
    try:
        stats_window.destroy()
    except:
        pass

    def get_top_raters():
        cur = connect_db()
        q = f"SELECT Author,COUNT(Dish) as Ratings FROM ratings GROUP BY
Author ORDER BY Ratings DESC LIMIT 5;"
        df = pd.read_sql_query(q,cur)
        return df

    raters_stat_window = show_window("Recipe Book")
    main_fr = show_frame(raters_stat_window)
    show_image(main_fr, "Recipebook.png")

```

```

fr_in= tk.Frame(master= main_fr,borderwidth=5,relief=tk.GROOVE,bg=
"#ffb5d1")
fr_in.pack(pady=70)
fr_box= tk.Frame(master= main_fr,relief=tk.GROOVE,bg=
"#ffb5d1",borderwidth=5,width= 60, height= 50)
fr_box.pack(pady=10)
fr_btn= tk.Frame(master= main_fr,bg= "#febebf")
fr_btn.pack(pady=10)

lbl_ch= tk.Label(master= fr_in, text="Top Raters", font=("Helvetica",
16),bg= "#ffb5d1")
lbl_ch.grid(row=0,column=0)

df = get_top_raters()
fig= plt.figure(figsize=(6,5))
ax= fig.add_subplot(111)
ax.bar(df["Author"],df["Ratings"],color= "pink",edgecolor=
"black",width=0.5)
ax.set_xlabel("Username", fontdict= {"fontsize": 10, "fontweight":
"bold"})
ax.set_ylabel("Dishes Rated", fontdict= {"fontsize": 10, "fontweight":
"bold"})
ax.set_title("Top Raters", fontsize= 12, fontweight= "bold", color=
"lightcoral")

canvas= FigureCanvasTkAgg(fig,master= fr_box)
canvas_widget= canvas.get_tk_widget()
canvas_widget.pack(side=tk.TOP,fill=tk.BOTH,expand=True, padx= 10, pady=
10)
canvas.draw()

btn_back= tk.Button(master= fr_btn, text="Back", relief= tk.RAISED,
borderwidth=5,bg= "#ffb5d1", font=("Helvetica",14), command= lambda:
show_stats_window(username))
btn_back.grid(row= 0,column=0)

raters_stat_window.mainloop()

# Start by creating database and tables if they don't exist
create_db()
show_welcome_window()

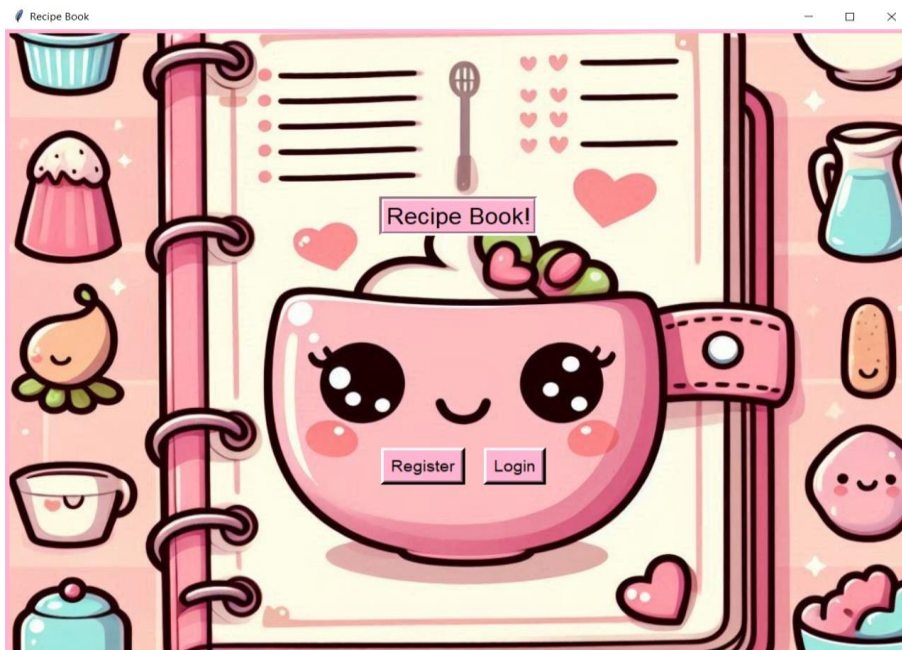
```

4.2 Github Repository

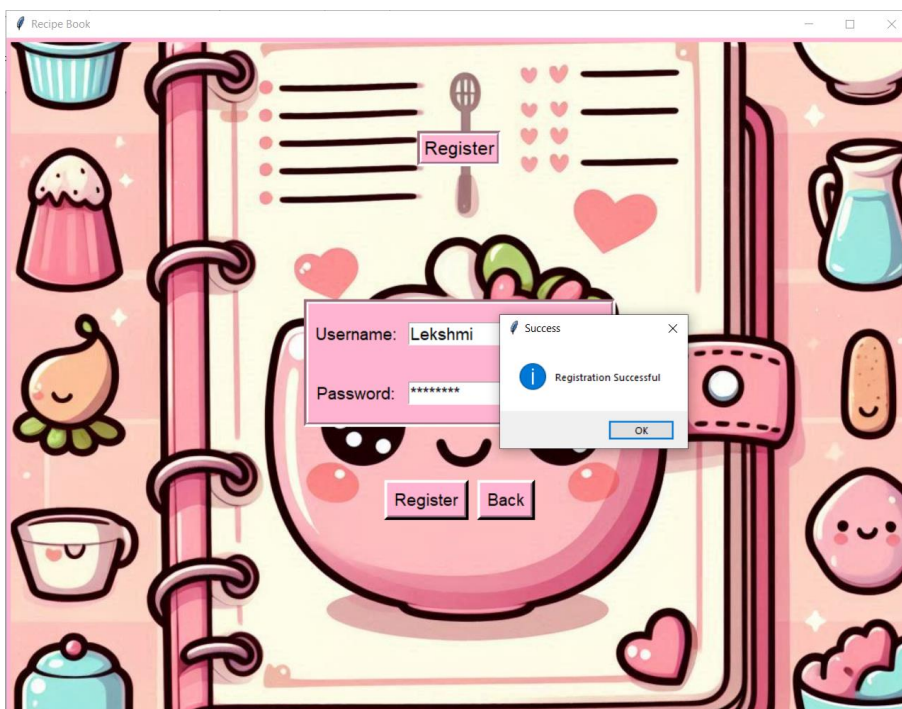
This code is available at: [Recipe-Book](#)

5. EXAMPLE OUTPUTS

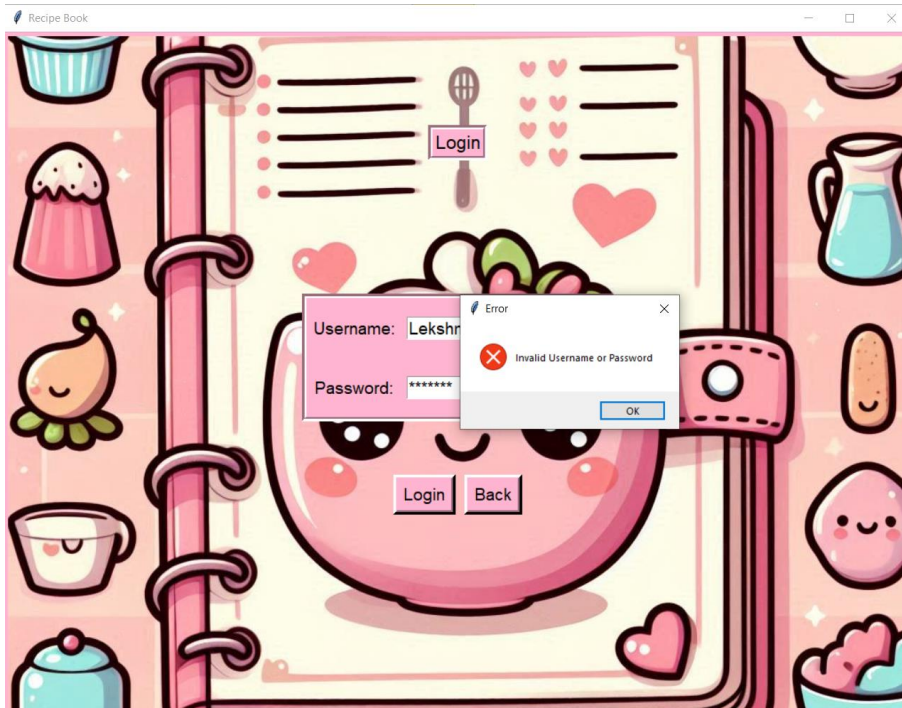
Welcome Screen



Registration



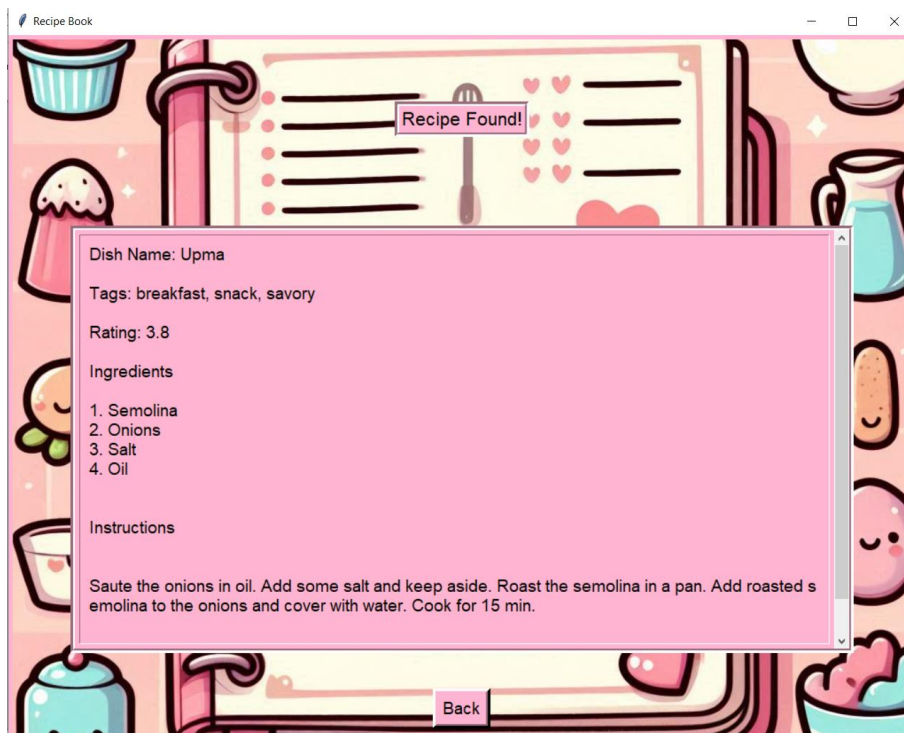
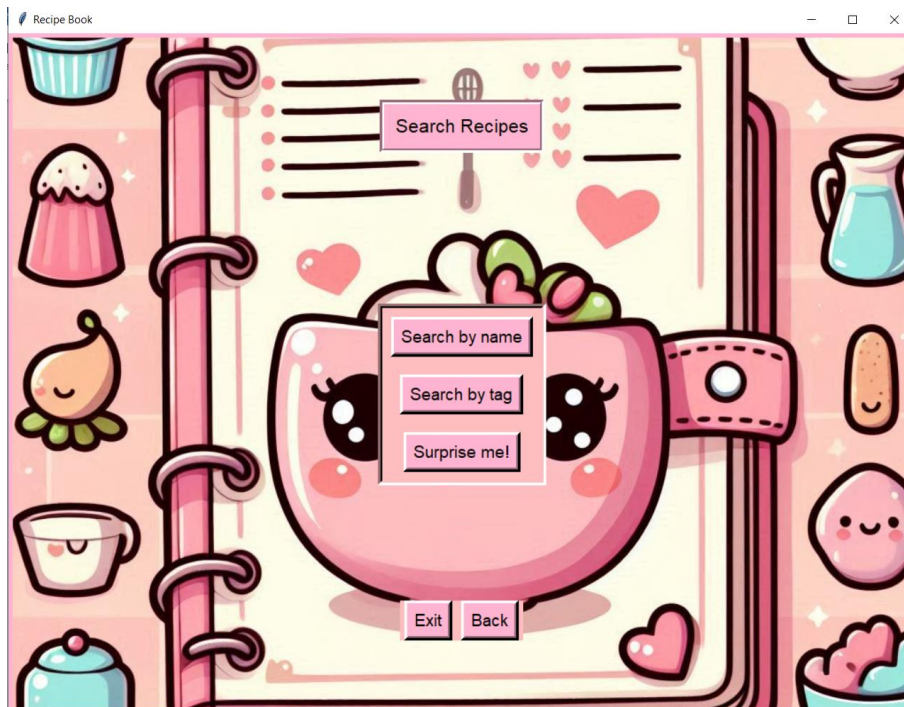
Login Validation

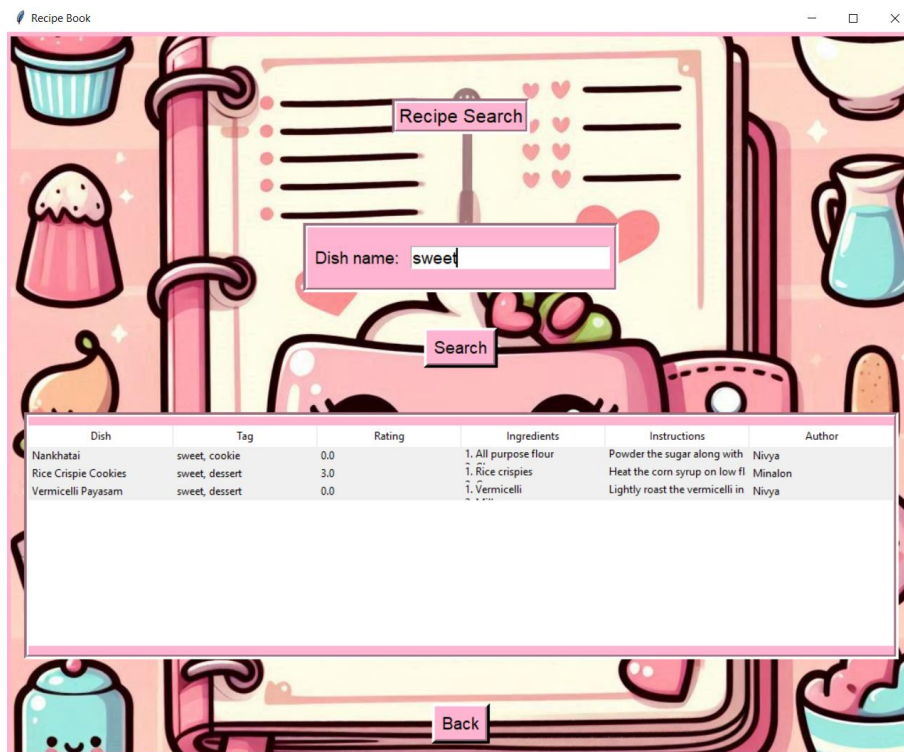


Home Screen

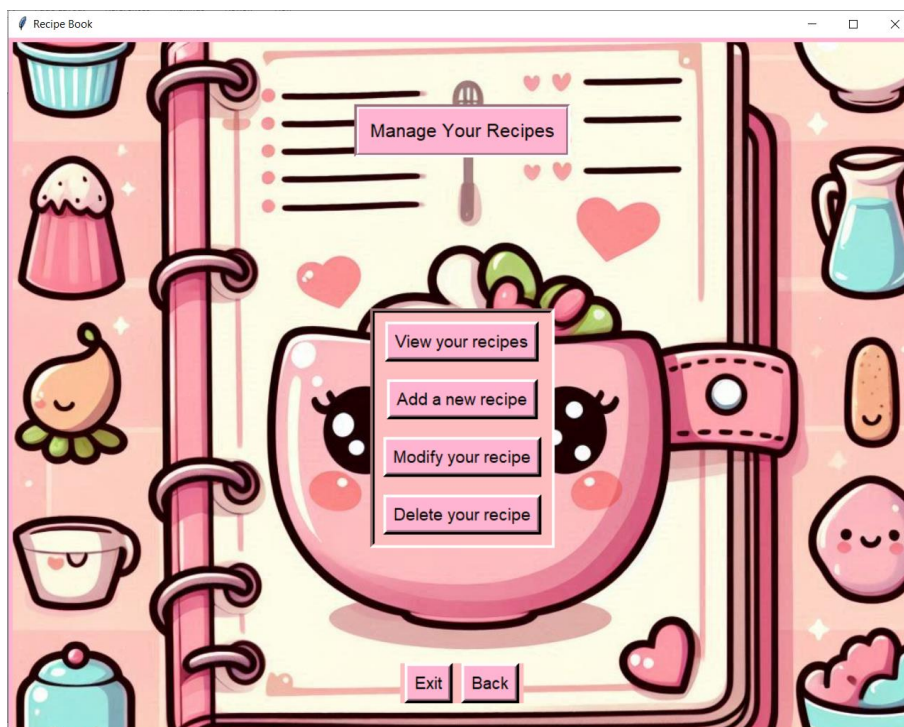


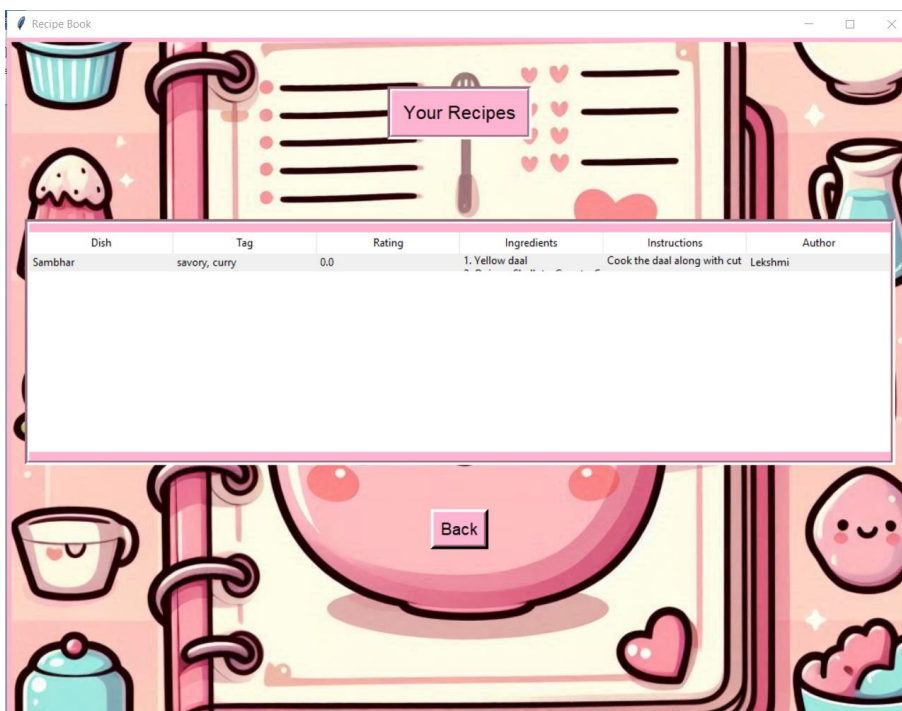
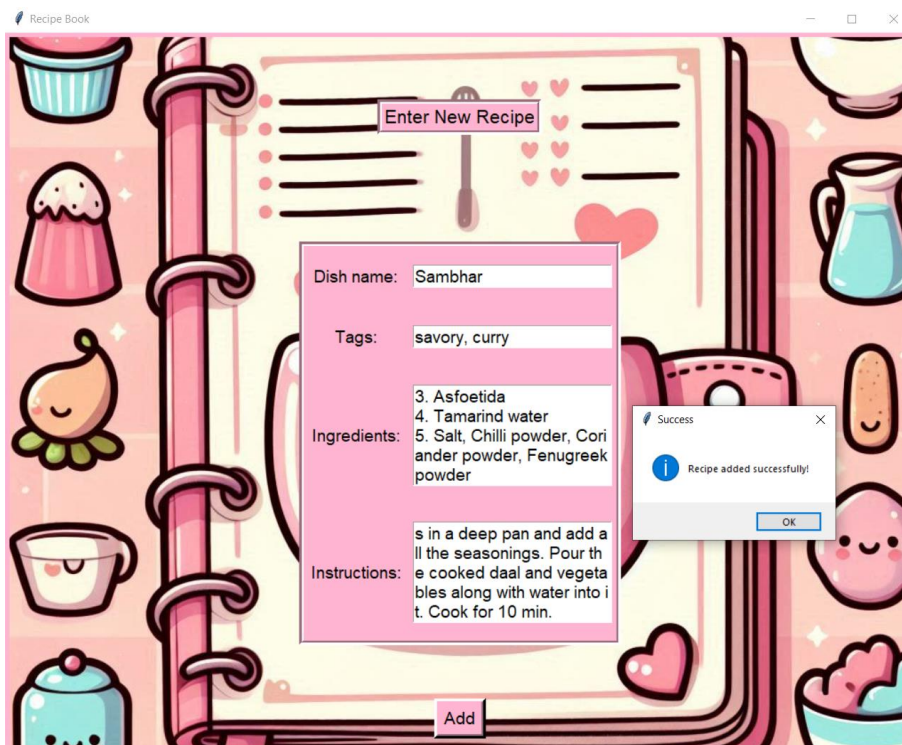
Search Recipes



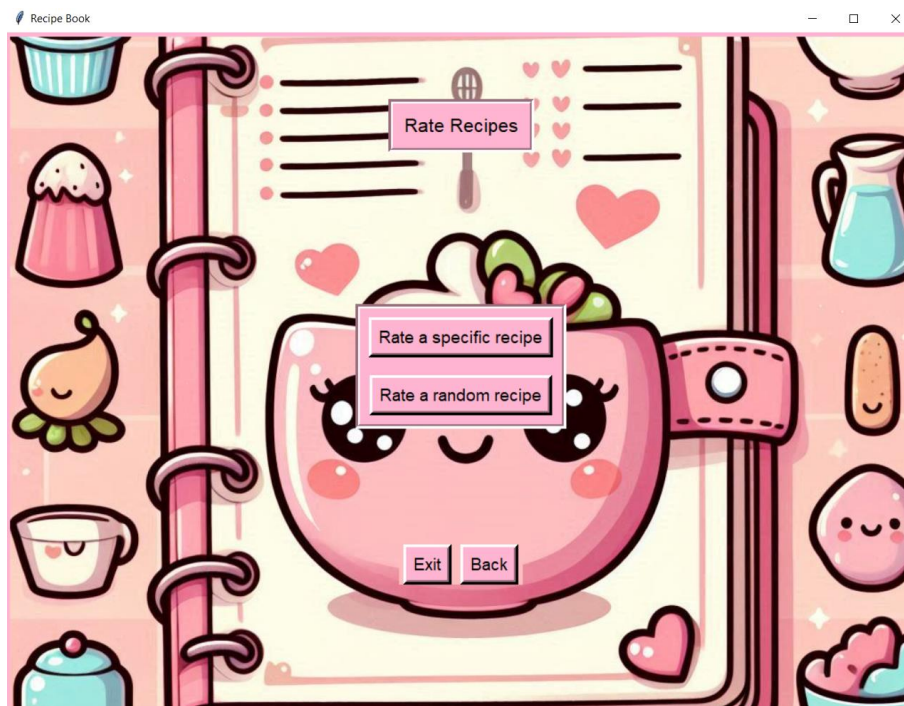
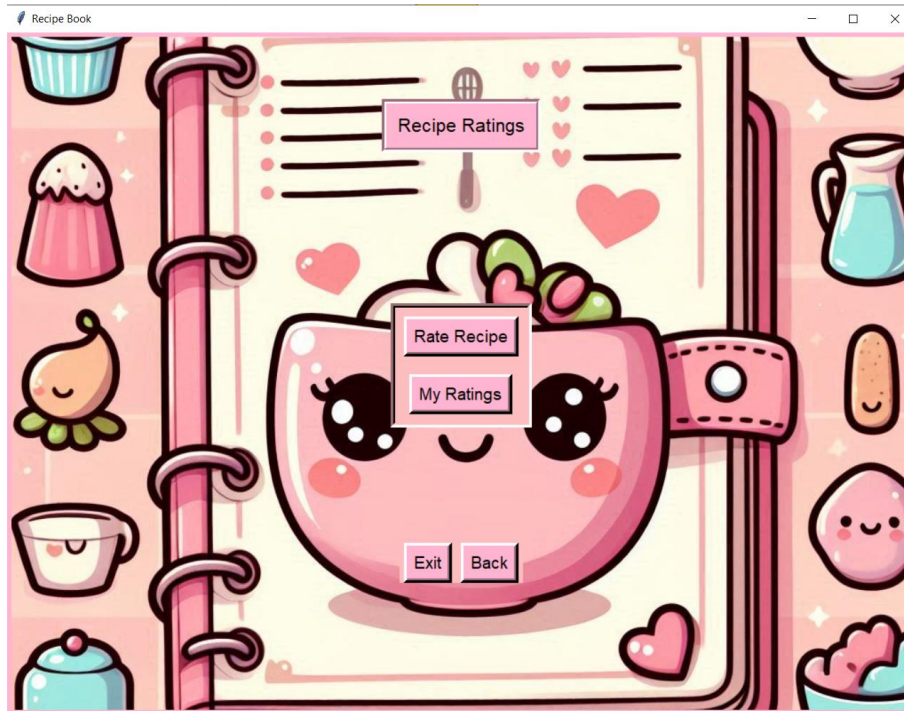


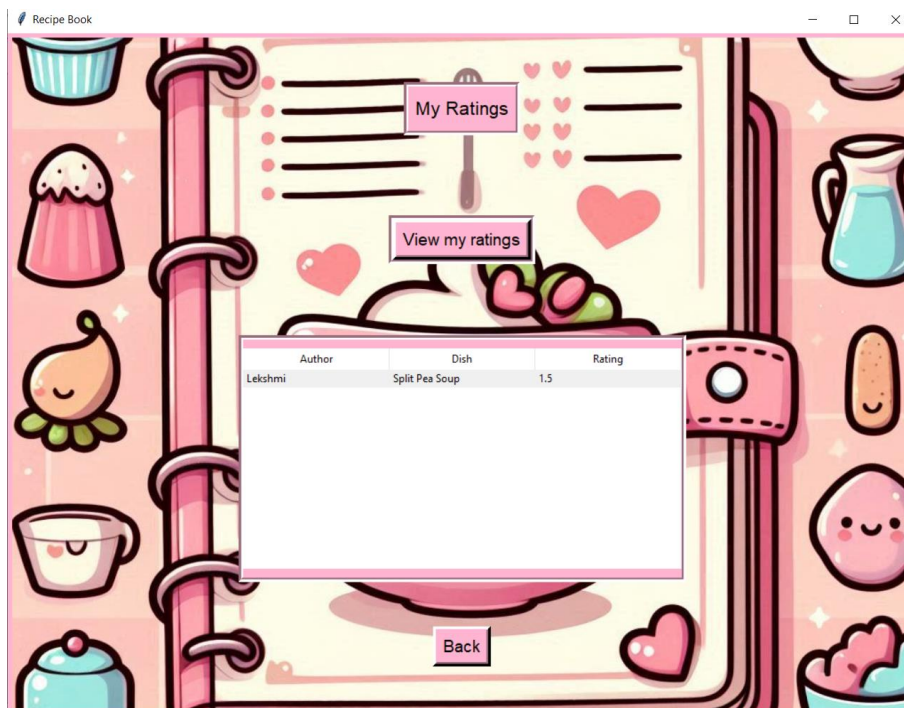
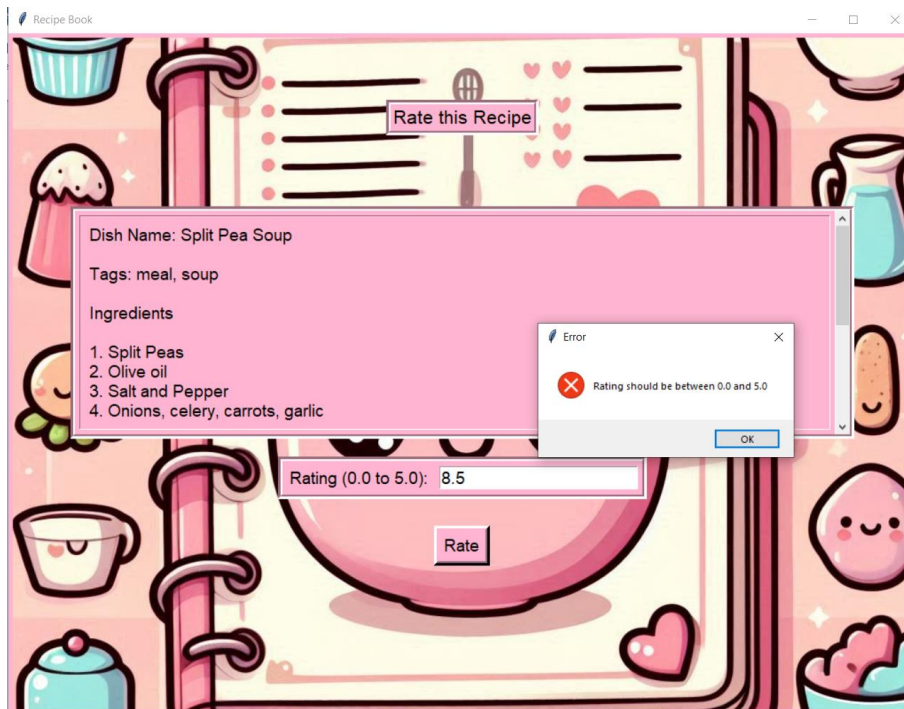
Manage Recipes



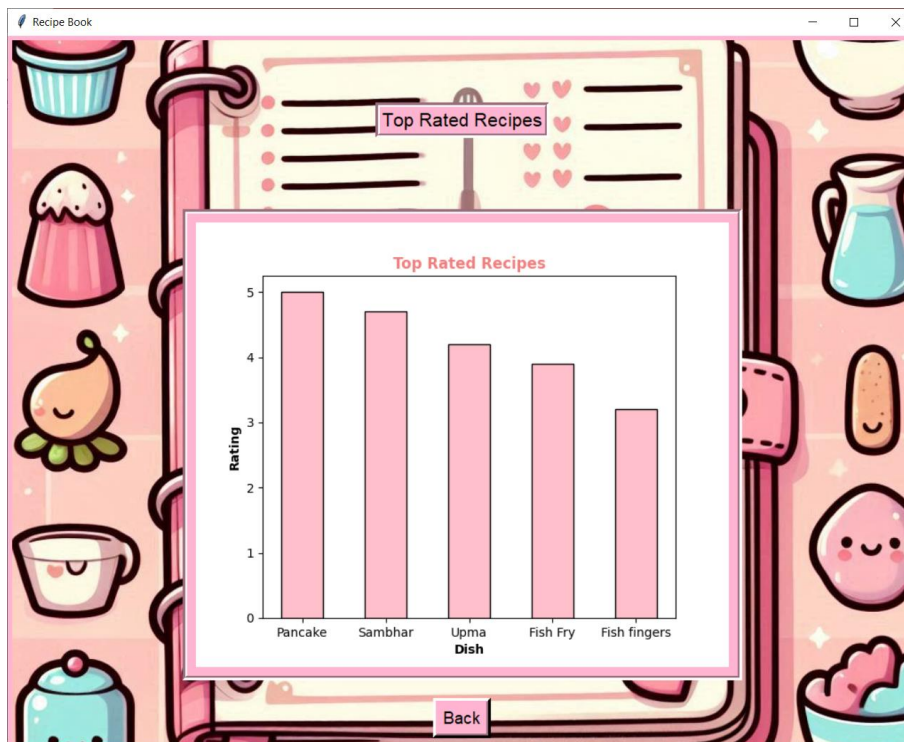
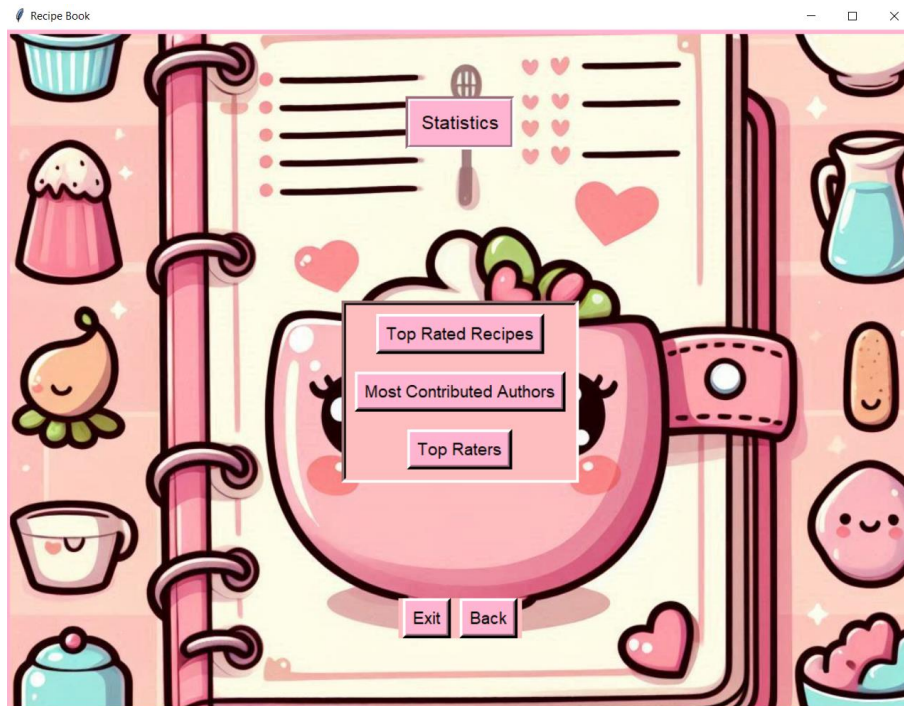


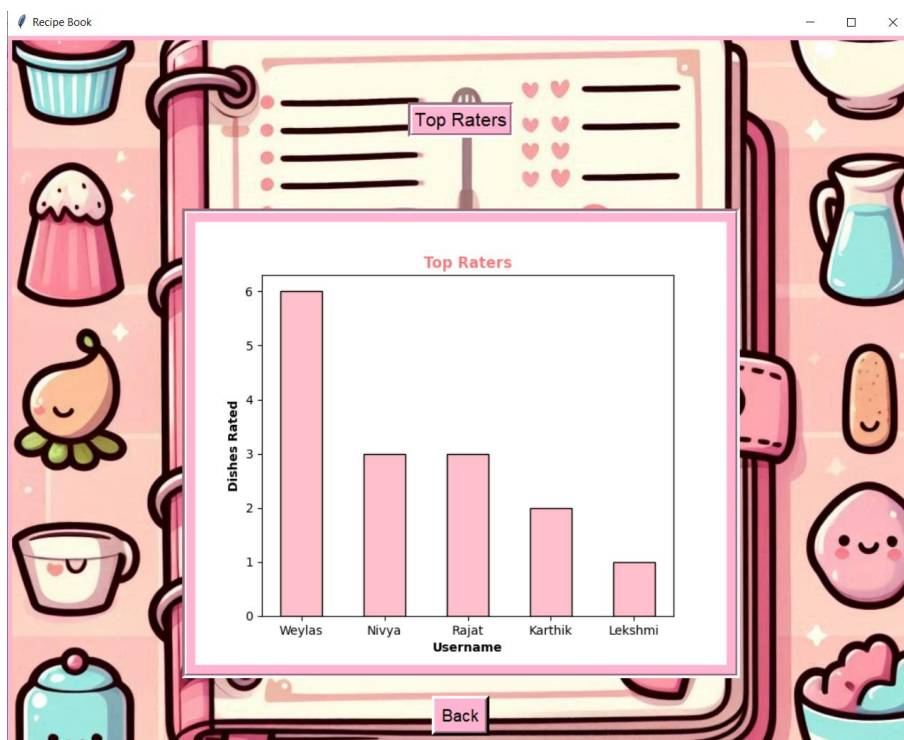
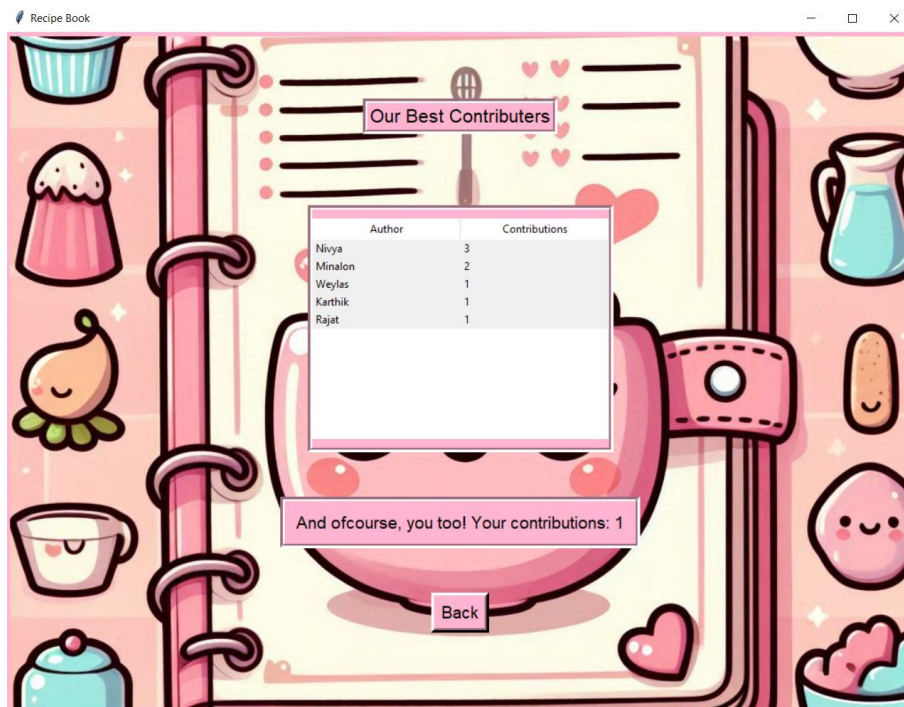
Rate Recipes



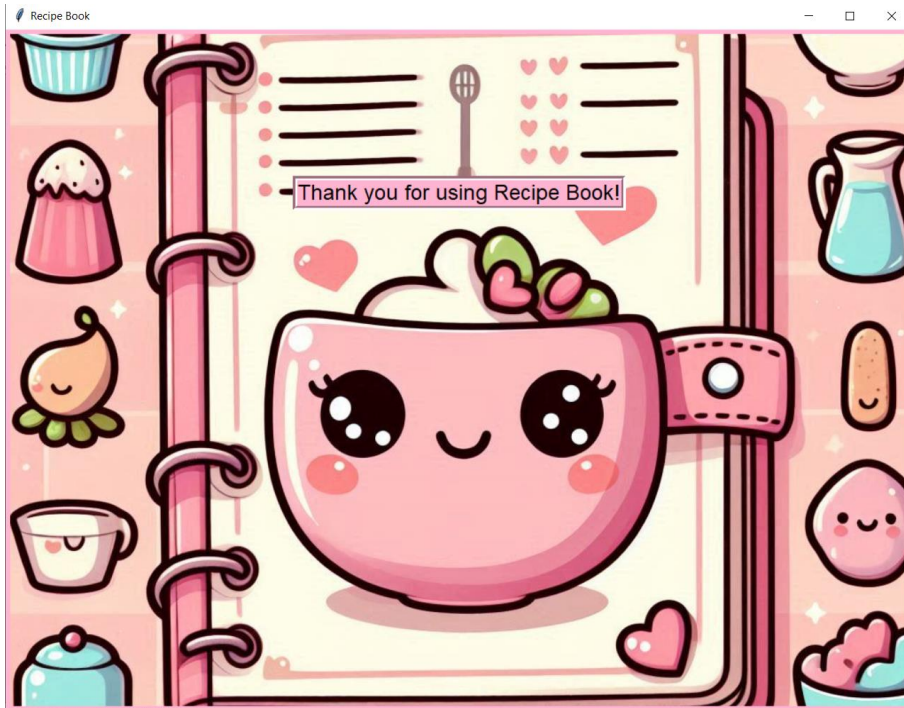


Recipe Book Stats





Exit Program



6. CONCLUSION

The program Recipe Book achieves all the intended functionalities with a neat and easy to use GUI. The program demonstrates the use of python libraries such as tkinter for GUI interface, matplotlib.pyplot for plotting graphs and mysql.connector for database connectivity with MySQL. Moreover, the program shows basic python code involving defining functions, argument passing, variable assignments, conditional statements and mathematical operations. The program thus meets all the requirements for a final project as outlined by the guidelines of Certificate Course in Coding Skills.

7. FUTURE SCOPE

This Recipe Book program can be improved in the future in both functionality as well as coding.

Some suggested functionality improvements are:

- i. Catching more database errors such as variable overflows and echoing it to the screen using message boxes. Right now only most basic errors are caught, so there is scope for a thorough testing and implementing more exception handling.
- ii. Adding any new functions based on user requirements such as a separate login for administrators to manage the whole application via the application itself. As of now, everyone is treated as a user.

Some coding improvements are:

- i. Cleaning up code to make it more user-friendly and easier to modify.
- ii. Bundling up repetitive code into functions to reduce program length. While some of the code is already revised to create functions for repetitive code, more can be done.

8. REFERENCES

1. NCERT Class 11 Informatics
2. NCERT Class 12 Informatics
3. <https://realpython.com/python-gui-tkinter/>
4. <https://docs.python.org/3/library/tkinter.html>
5. https://matplotlib.org/stable/gallery/user_interfaces/
6. <https://stackoverflow.com/questions>