# header information

"""

author: Maurice Farr

date: 12/9/2024

assignment: SDEV-140 GUI Tkinter App Final Project, Part 2

purpose: This .py file has the rest of the necessary functions to execute the GUI Tkinter App,

"My Glucose Meter Tracker".

"""

#ProjectFunctions.py

**#Source Code of All files (.py):**

from tkinter import\*

from tkinter import Message

import time

import pandas as pd

#Define confirmation message for the buttons (create\_account\_button and login\_button)

cnca\_message1 = "ERROR: Please fill out the required information to complete this process"

cnca\_message2 = " Congrats, you have officially created an account. You can now use your created account to login."

login\_message1= "Sorry, you were unable to login."

login\_message2 = " Successfully logged in."

#Path for file for to store and retrieve user data

path\_for\_file = "C:\\Users\\artvs\\OneDrive\\Documents\\SDEV 140 Final Project\\MauriceFarr-DatabaseForFinalProject.csv"

#Define the function to destroy a window

def destroyWindow(widget):

widget.destroy()

#Define the function and add grid layouts for creating a window when clicking the new account button

def new\_account\_button\_clicked():

window\_n\_a = Toplevel()

window\_n\_a.geometry("600x500")

window\_n\_a.config(bg="#ffffff")

#Display the title name of the app on the "new\_account\_button\_clicked" window including adding its grid layouts

title\_label2 = Label(window\_n\_a, text = "My Glucose Meter Tracker", font=("Californian FB", 20, "bold"), fg="#1197e0",bg="#ffffff")

title\_label2.grid(row=5, column=5, padx=30, pady=30)

#Create the label and add grid layouts for the email

email\_label = Label(window\_n\_a, text = "Email", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

email\_label.grid(row=20, column=3, padx=20, pady=20)

#Create an entry text field and add grid layouts for the email

entry\_email1 = Entry(window\_n\_a, width=40)

entry\_email1.grid(row=20, column=5, padx=30, pady=30)

entry\_email1.config(bg="#ffffff")

#Create the label and add grid layouts for the username

username\_label = Label(window\_n\_a, text = "Username", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

username\_label.grid(row=24, column=3, padx=20, pady=20)

#Create an entry text field and add grid layouts for the username

entry\_username = Entry(window\_n\_a, width=40)

entry\_username.grid(row=24, column=5, padx=30, pady=30)

entry\_username.config(bg="#ffffff")

#Create the label and add grid layouts for the password

password\_label = Label(window\_n\_a, text = "Password", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

password\_label.grid(row=28, column=3, padx=20, pady=20)

#Create an entry text field and add grid layouts for the password

entry\_password1 = Entry(window\_n\_a, width=40)

entry\_password1.grid(row=28, column=5, padx=30, pady=30)

entry\_password1.config(bg="#ffffff")

#Apply a "create new account" button including adding its grid layouts

create\_account\_button = Button(window\_n\_a, text = "Create New Account", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff", command= lambda:display\_confirmation\_message(window\_n\_a, entry\_email1, entry\_username, entry\_password1, cnca\_message1, cnca\_message2))

create\_account\_button.grid(row=31, column=5, columnspan=3, padx=30, pady=20)

#Create a back button and add grid layouts for the "new\_account\_button\_clicked" window

back\_button1 = Button(window\_n\_a, text="Back", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff", command=lambda: destroyWindow(window\_n\_a))

back\_button1.grid(row=1, column=0, padx=30, pady=20)

#Define the function to open a new window to display the message that new account has been created including adding grid layouts and the parameters

def display\_confirmation\_message(window\_combine, entry\_email1, entry\_username, entry\_password1, display\_message1, display\_message2):

email = entry\_email1.get()

username = entry\_username.get()

password = entry\_password1.get()

#Conditional if-else statement to check if there are any entries filled in the entry box for creating new account

if len(email) != 0 and len(username) != 0 and len(password) != 0: #Indication that no entries are filled out in the entry box

window\_cnca = Toplevel()

window\_cnca.geometry("700x65")

window\_cnca.config(bg="#ffffff")

cnca\_label = Label(window\_cnca, text= display\_message2, font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

cnca\_label.grid(row=28, column=3, padx=20, pady=20)

window\_cnca.after(3000, lambda: destroyWindow(window\_cnca)) # Indication that the cnca\_message1 will display for 3 seconds before disappearing

window\_combine.destroy() # Both windows (window\_n\_a and window\_cnca) will disappear

else:

window\_cnca = Toplevel()

window\_cnca.geometry("700x65")

window\_cnca.config(bg="#ffffff")

cnca\_label = Label(window\_cnca, text= display\_message1, font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

cnca\_label.grid(row=28, column=3, padx=20, pady=20)

window\_cnca.after(3000, lambda: destroyWindow(window\_cnca)) # Indication that the cnca\_message2 will display for 3 seconds before disappearing

window\_combine.destroy() # Both windows (window\_n\_a and window\_cnca) will disappear

#Define the function to open a new window when clicking the login button including adding its grid layouts

def login\_button\_clicked():

window\_login = Toplevel()

window\_login.geometry("700x450")

window\_login.config(bg="#ffffff")

#Display the title name of the app on the "login\_button\_clicked" window including adding its grid layouts

title\_label2 = Label(window\_login, text = "My Glucose Meter Tracker", font=("Californian FB", 20, "bold"), fg="#1197e0",bg="#ffffff")

title\_label2.grid(row=5, column=5, padx=30, pady=30)

#Create the label and add grid layouts for the email

username\_email\_label = Label(window\_login, text = "Username or Email", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

username\_email\_label.grid(row=20, column=3, padx=20, pady=20)

#Create the entry text field and add grid layouts for the mail

entry\_username\_email = Entry(window\_login, width=40)

entry\_username\_email.grid(row=20, column=5, padx=30, pady=30)

entry\_username\_email.config(bg="#ffffff")

#Create the label and add grid layouts for the password

password\_label = Label(window\_login, text = "Password", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

password\_label.grid(row=28, column=3, padx=20, pady=20)

#Create the entry text field and add grid layouts for the password

entry\_password2 = Entry(window\_login, width=40)

entry\_password2.grid(row=28, column=5, padx=30, pady=30)

entry\_password2.config(bg="#ffffff")

#Create a login button including adding its grid layouts

login\_button = Button(window\_login, text = "Login", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff", command=lambda: login\_process(entry\_username\_email, entry\_password2, window\_login))

login\_button.grid(row=31, column=5, columnspan=3, padx=30, pady=20)

#Create a back button for the "login\_button\_clicked" window including adding its grid layouts

back\_button2 = Button(window\_login, text="Back", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff", command= lambda: destroyWindow(window\_login))

back\_button2.grid(row=1, column=0, padx=30, pady=20)

#Define the function for the login process by including parameters (entry box for both username and password, and window\_login) and the grid layouts

def login\_process(username\_entrybox, password\_entrybox, window\_login):

username= username\_entrybox.get()

password= password\_entrybox.get()

#Import user data by reading the file using the cvs method

df1 = pd.read\_csv ("C:\\Users\\artvs\\OneDrive\\Documents\\SDEV 140 Final Project\\MauriceFarr-DatabaseForFinalProject.csv")

#Identify if a character exist in username by connecting to the database

if '@' in username:

entries = df1[(df1["email"] == username) & (df1["password"] == password)]

else:

entries = df1[(df1["username"]==username) & (df1["password"]==password)]

#Conditional if-else statement to check if whether there are any entries after clicking the login button

if len(entries) == 0: #indication that there are missing or invalid info into entry text field when logging in

window\_error1 = Toplevel()

window\_error1.geometry("300x65")

window\_error1.config(bg="#ffffff")

error1\_label = Label(window\_error1, text= login\_message1, font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

error1\_label.grid(row=28, column=3, padx=20, pady=20)

window\_error1.after(3000, lambda: destroyWindow(window\_error1)) # Indication that the login\_message1 will display for 3 seconds before disappearing

window\_login.destroy() # Window\_login will disappear after failing to login

else:

#Indication that the user was able to log in

window\_login\_succces = Toplevel()

window\_login\_succces.geometry("220x65")

window\_login\_succces.config(bg="#ffffff")

window\_login\_succces\_label = Label(window\_login\_succces, text=login\_message2, font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

window\_login\_succces\_label.grid(row=28, column=3, padx=20, pady=20)

window\_login\_succces.after(3000, lambda: destroyWindow(window\_login\_succces)) # Indication that the login\_message2 will display for 3 seconds before disappearing

window\_login.destroy() # Window\_login will disappear after logging

display\_blood\_glucose\_level() #Transition to open a new window,"window\_blood\_glucose\_level", after logging in

#Define the function to create window for the blood glucose level including adding grid layouts

def display\_blood\_glucose\_level():

window\_blood\_glucose\_level = Toplevel()

window\_blood\_glucose\_level.geometry("600x250")

window\_blood\_glucose\_level.config(bg="#ffffff")

#Display the title label and add grid layouts for the "blood\_glucose\_level" window

title\_label3 = Label(window\_blood\_glucose\_level, text = "Please enter your blood glucose level", font=("Californian FB", 18, "bold"), fg="#1197e0",bg="#ffffff")

title\_label3.grid(row=5, column=2, padx=(140,10), pady=30)

#Create entry text field and add grid layouts for blood glucose level

entry\_blood\_glucose\_level = Entry(window\_blood\_glucose\_level, width=20)

entry\_blood\_glucose\_level.grid(row=15, column=2, padx=(140,20), pady=20)

entry\_blood\_glucose\_level.config(bg="#ffffff")

#Display the title label of the measurement units, milligrams (mg) per decilitre (dL), including adding its grid layouts

title\_label4 = Label(window\_blood\_glucose\_level, text="mg/dL",font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

title\_label4.grid(row=15, column=2, padx=(350,20), pady=20)

#Apply a "submit" button for the blood glucose level including adding its grid layouts

submit\_button = Button(window\_blood\_glucose\_level, text="Submit", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff", command= lambda: submit\_blood\_glucose (window\_blood\_glucose\_level,entry\_blood\_glucose\_level.get()))

submit\_button.grid(row=20, column=2, columnspan=1, padx=(140,10), pady=20)

#Define the function for the results when the user submit their blood glucose level including adding its parameters (window\_combine, user\_glucose) and grid layouts

def submit\_blood\_glucose(window\_combine, user\_glucose):

if len(user\_glucose) == 0: #Check to see if there are any empty entries for entering blood glucose level

window\_error2 = Toplevel()

window\_error2.geometry("490x60")

window\_error2.config(bg="#ffffff")

error2\_label = Label(window\_error2, text=" ERROR: Please enter your blood glucose level!",font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

error2\_label.grid(row=28, column=3, padx=20, pady=20)

window\_error2.after(3000, lambda: destroyWindow(window\_error2)) #The "error\_2" message will display for 3 seconds before disappearing

else:

user\_glucose = float(user\_glucose) #Indication we can now enter a number

if user\_glucose < 0: #Determine if blood glucose level is less than zero

window\_error3 = Toplevel()

window\_error3.geometry("455x60")

window\_error3.config(bg="#ffffff")

error3\_label = Label(window\_error3, text=" ERROR: You cannot have a negative value!", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

error3\_label.grid(row=28, column=3, padx=20, pady=20)

window\_error3.after(3000, lambda: destroyWindow(window\_error3)) #The "error\_3" message will display for 3 seconds before disappearing

elif user\_glucose != int(user\_glucose): #Determine if blood glucose level is a decimal number

window\_error4 = Toplevel()

window\_error4.geometry("455x60")

window\_error4.config(bg="#ffffff")

error4\_label = Label(window\_error4, text=" ERROR: You cannot have a decimal value!", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

error4\_label.grid(row=28, column=3, padx=20, pady=20)

window\_error4.after(3000, lambda: destroyWindow(window\_error4)) #The "error\_4" message will display for 3 seconds before disappearing

else:

if user\_glucose >150: #The range for target numbers for high blood glucose levels

window\_high = Toplevel()

window\_high.geometry("250x60")

window\_high.config(bg="#ffffff")

high\_label = Label(window\_high, text=" Blood glucose is TOO HIGH.", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

high\_label.grid(row=28, column=3, padx=20, pady=20)

window\_high.after(3000, lambda: destroyWindow(window\_high)) #The high message will display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_high and window\_blood\_glucose\_level) will disappear

elif user\_glucose >= 100 and user\_glucose <= 150: #The range for target numbers for normal blood glucose levels

window\_normal = Toplevel()

window\_normal.geometry("250x60")

window\_normal.config(bg="#ffffff")

normal\_label = Label(window\_normal, text=" Blood glucose is NORMAL.", font=("Californian FB", 12, "bold"),fg="#1197e0", bg="#ffffff")

normal\_label.grid(row=28, column=3, padx=20, pady=20)

window\_normal.after(3000, lambda: destroyWindow(window\_normal)) #The "normal" message will display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_normal and window\_blood\_glucose\_level) will

elif user\_glucose >= 71 and user\_glucose <= 99: #The range target numbers for borderline low blood glucose levels

window\_borderline = Toplevel()

window\_borderline.geometry("330x60")

window\_borderline.config(bg="#ffffff")

borderline\_label = Label(window\_borderline, text=" Blood glucose is BORDERLINE LOW.", font=("Californian FB", 12, "bold"),fg="#1197e0", bg="#ffffff")

borderline\_label.grid(row=28, column=3, padx=20, pady=20)

window\_borderline.after(3000, lambda: destroyWindow(window\_borderline)) #The "borderline low" message will display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_borderline and window\_blood\_glucose\_level) will disappear

elif user\_glucose <= 70: #The range for target numbers for low blood glucose levels

window\_low = Toplevel()

window\_low.geometry("230x60")

window\_low.config(bg="#ffffff")

low\_label = Label(window\_low, text=" Blood glucose is LOW.", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

low\_label.grid(row=28, column=3, padx=20, pady=20)

window\_low.after(3000, lambda: destroyWindow(window\_low)) #The "low" message display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_low and window\_blood\_glucose\_level) will disappear

#Define the function for clicking the reset new password button including adding grid layouts

def forgot\_reset\_password\_button\_clicked():

window\_frp = Toplevel()

window\_frp.geometry("750x520")

window\_frp.config(bg="#ffffff")

#Display the title name of the app on the "forgot\_reset\_password\_clicked" window including adding its grid layouts

title\_label2 = Label(window\_frp, text = "My Glucose Meter Tracker", font=("Californian FB", 20, "bold"), fg="#1197e0",bg="#ffffff")

title\_label2.grid(row=5, column=2, columnspan=1, padx=30, pady=40)

#Create the label and add grid layouts for the email

email\_label2 = Label(window\_frp, text="Email", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

email\_label2.grid(row=20, column=1, columnspan=1, padx=20, pady=20)

#Create the entry text field and add grid layouts for the email

entry\_email2= Entry(window\_frp, width=40)

entry\_email2.grid(row=20, column=2, padx=30, pady=30)

entry\_email2.config(bg="#ffffff")

#Create the label and add grid layouts to reset the password

reset\_password\_label = Label(window\_frp, text = "Type New Password", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

reset\_password\_label.grid(row=23, column=1, columnspan=1, padx=20, pady=20)

#Create the entry text field and add grid layouts to type the new password

entry\_password3=Entry(window\_frp, width=40)

entry\_password3.grid(row=23, column=2, padx=30, pady=30)

entry\_password3.config(bg="#ffffff")

#Create the label and add grid layouts to confirm new the password

confirm\_new\_password\_label = Label(window\_frp, text = "Retype New Password", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff")

confirm\_new\_password\_label.grid(row=26, column=1, columnspan=1, padx=20, pady=20)

#Create the entry text field and add grid layouts to confirm new password including

entry\_confirm\_new\_password = Entry(window\_frp, width=40)

entry\_confirm\_new\_password.grid(row=26, column=2, padx=30, pady=30)

entry\_confirm\_new\_password.config(bg="#ffffff")

#Apply a "create new password" button including adding its grid layouts

create\_new\_password\_button = Button(window\_frp, text = "Create New Password", font=("Californian FB", 12, "bold"), fg="#1197e0",bg="#ffffff", command= lambda: update\_password(window\_frp, path\_for\_file, entry\_email2, entry\_password3,entry\_confirm\_new\_password))

create\_new\_password\_button.grid(row=27, column=2, columnspan=5, padx=30, pady=20)

#Create a back button for the "forgot\_reset\_password\_button\_clicked" window including adding its grid layouts

back\_button3 = Button(window\_frp, text="Back", font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff",command=lambda: destroyWindow(window\_frp))

back\_button3.grid(row=1, column=0, padx=30, pady=20)

#Define the function to reset new password using parameters (email, password, and retype password) for the entry box by connecting to the database

def update\_password(window\_combine, path\_for\_file, email\_entrybox, new\_password\_entrybox, retype\_new\_password\_entrybox):

email = email\_entrybox.get() #Get the user to enter their email in the entrybox

new\_password = new\_password\_entrybox.get() ##Get the user to enter new password in the entrybox

retype\_new\_password = retype\_new\_password\_entrybox.get() #Get the user to reenter their new password in the entrybox for confirmation

#Import user data by reading the file using the cvs method

df1 = pd.read\_csv("C:\\Users\\artvs\\OneDrive\\Documents\\SDEV 140 Final Project\\MauriceFarr-DatabaseForFinalProject.csv")

#Conditional if-else statements for resetting new password

if (len (new\_password) != 0 #Check if there is an entry to type new password

and len (retype\_new\_password) != 0 #Check if there is an entry to retype new password for confirmation

and new\_password == retype\_new\_password): #Check if the password that is typed in both entrybox (the new\_password and retype\_new\_password) matches with each other

#Identify if a character exist in email by connecting to the database

df1.loc[df1["email"] == email, 'password'] = retype\_new\_password

df1.to\_csv(path\_for\_file, index=False)

#Create a window and adding grid layouts to display the message that the new password has been created

window\_npw = Toplevel()

window\_npw.geometry("600x65")

window\_npw.config(bg="#ffffff")

pw\_label = Label(window\_npw,text=" New password has been reset. You can now use your new password to login.",font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

pw\_label.grid(row=28, column=3, padx=20, pady=20)

window\_npw.after(3000, lambda: destroyWindow(window\_npw)) #The new password confirmation will display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_frp and window\_npw) will disappear

else: #Create a window and adding grid layouts to display an error message while creating new password

window\_npw = Toplevel()

window\_npw.geometry("330x65")

window\_npw.config(bg="#ffffff")

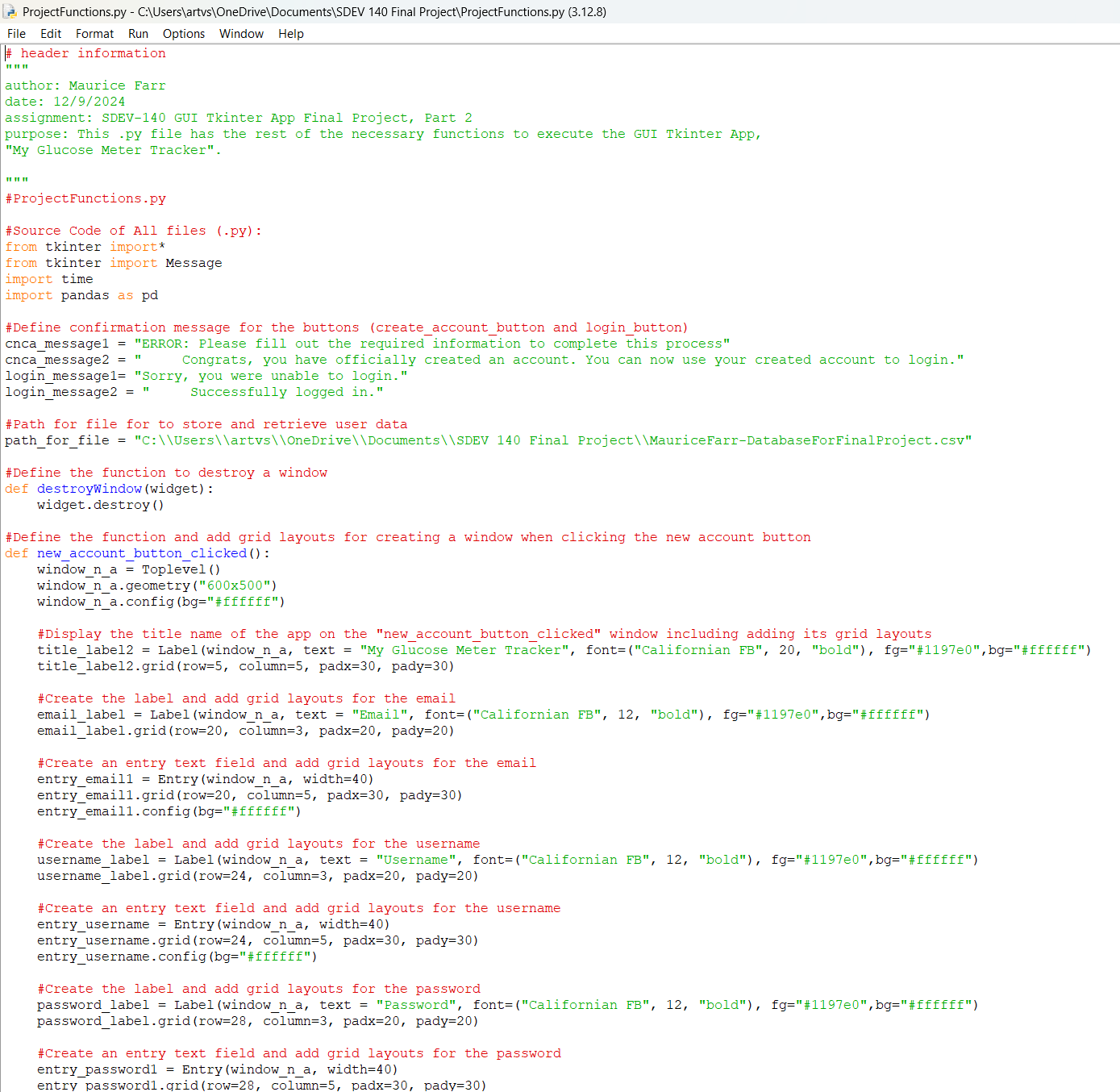
pw\_label = Label(window\_npw,text="ERROR: Please fill out or correct any entries!",font=("Californian FB", 12, "bold"), fg="#1197e0", bg="#ffffff")

pw\_label.grid(row=28, column=3, padx=20, pady=20)

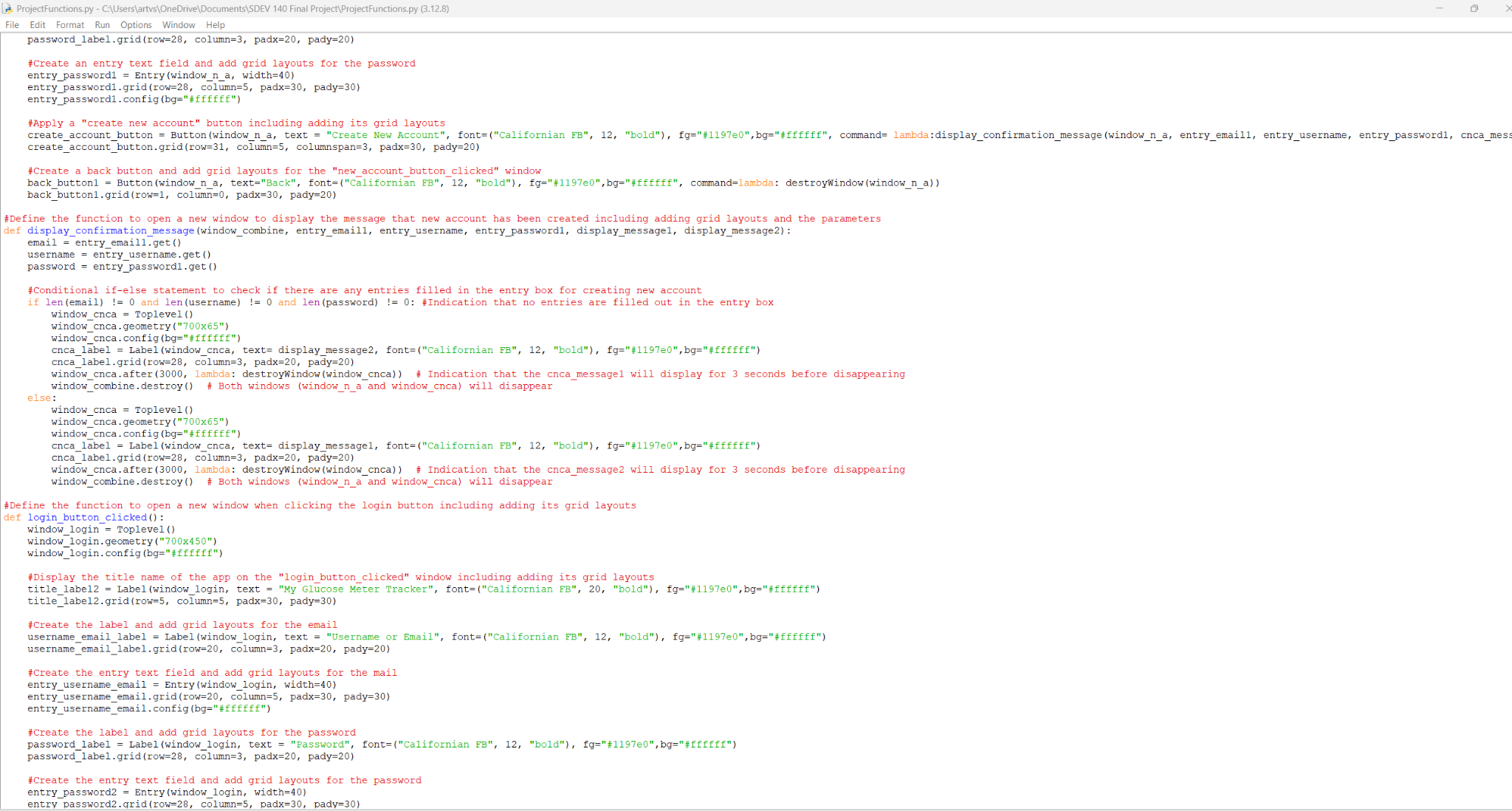
window\_npw.after(3000, lambda: destroyWindow(window\_npw)) #The error message will display for 3 seconds before disappearing

window\_combine.destroy() #Both windows (window\_frp and window\_npw) will disappear

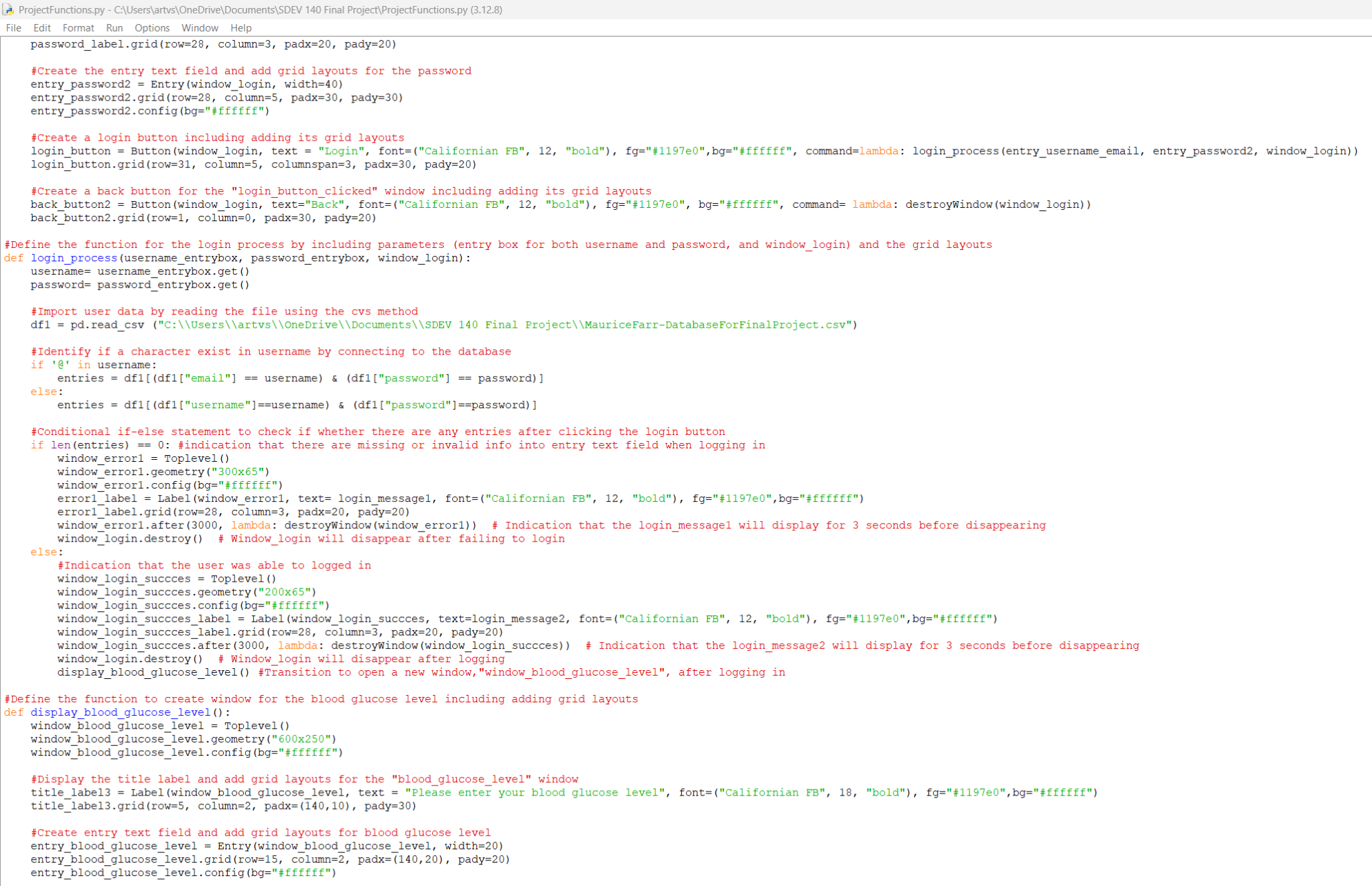
**Screenshot of Source Code of All Files #1:**



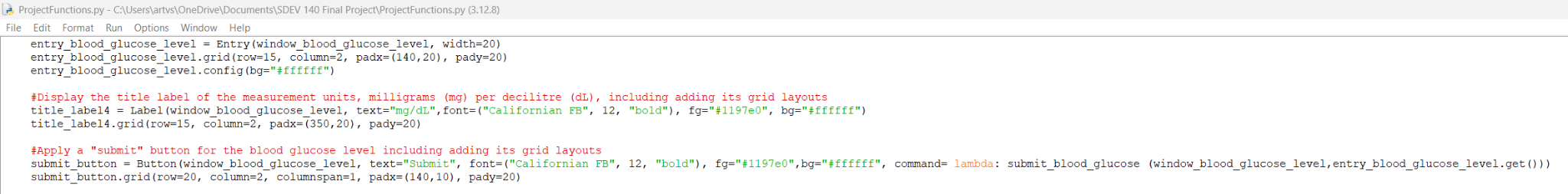
**Screenshot of Source Code of All Files #2:**

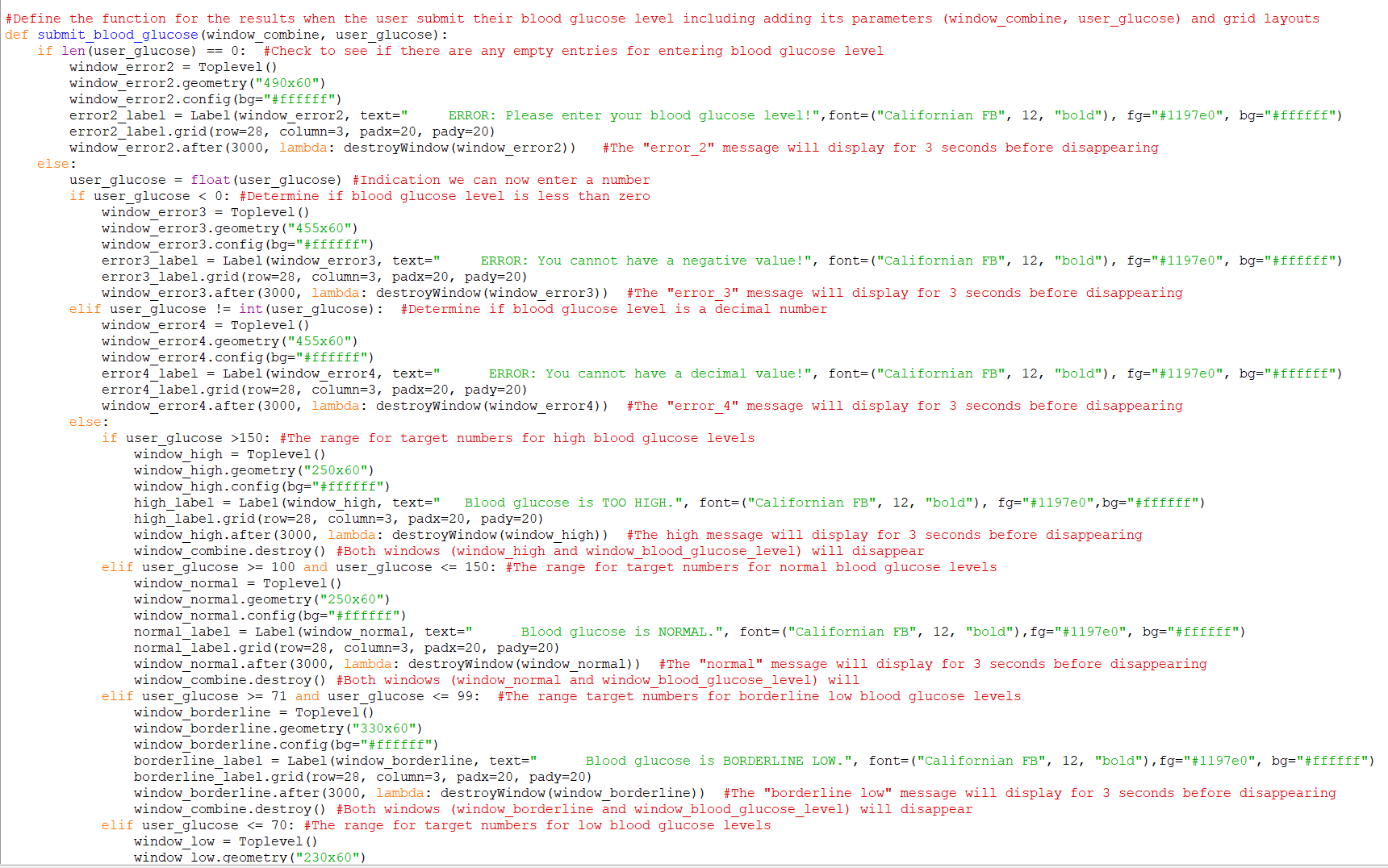
****

**Screenshot of Source Code of All Files #3:**

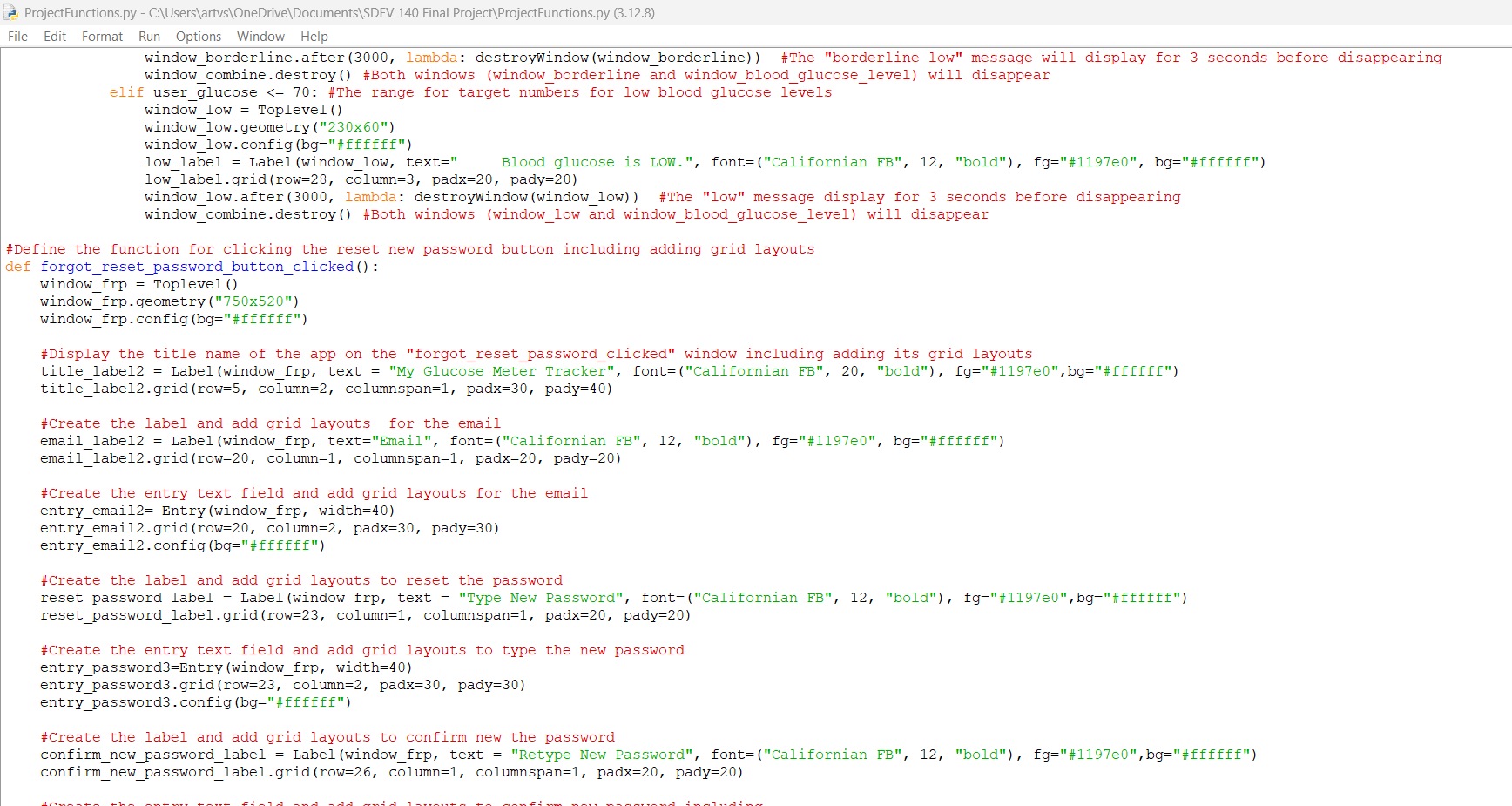
****

**Screenshot of Source Code of All Files #4:**

****

**Screenshot of Source Code of All Files #5:**

**Screenshot of Source Code of All Files #6:**



**Screenshot of Source Code of All Files #7:**



**Screenshot of Source Code of All Files #8:**

