d r á k o n

*personal budget tracker*

Arooba Siddiqi 321618

Emaan Bashir 296190

BESE-10A



We made a budget tracker, in the form of a desktop GUI, with the objective to help hostelites in managing their finances.

Initially we wanted to create a mobile application, but after doing a little research on kivy, tkinter etc, we decided to go with the simpler option due to time restrictions. It is important to note that deciding the topic and what modules to use was perhaps the most difficult part of the project.

We had watched a few tutorials on tkinter and headed straight into creating the account system, and the screen for it. While doing this, we realized that backend and frontend work would have to be divided in order to have an efficient work process.

After discussing what screens we need, I started creating screens and sending them to Emaan who would add the backend logic to buttons, labels and such on the screen. We were also identifying errors in the code and fixing them simultaneously. This also included any errors that the user had to be shown, e.g. if the user entered an alphabet while giving budget goals an error message would be shown to user.

There are 9 main screens :

* Launch Screen
* Register
* First budget goal entry
* Homepage
* Change Budget
* Change Password
* Daily spending
* Monthly Review
* Yearly Analysis

Below are some details of the workings of screens.

Launch, Register, & Change Password Screens

These screens handle the accounts. To create databases with unique accounts, we used a library called sqlite.

(Register Screen)

*Required Entry Fields :*

* First & Last name
* Username
* Password

*Errors :*

* Only alpha characters allowed
* All textboxes must have entries
* Username must be unique
* Password must be at least 8 characters
* Two entries of passwords must match

Budget Entry Screens

Another table related to the accounts table using the unique usernames, is created to deal with budget. Budget will be entered the first time you use the app, i.e. when you register yourself, and if ever you need to change the budget.

*Required Entry Fields :*

* Monthly Budget
* Tuition & Living Expense

[Budget goals]

* Food
* Entertainment
* Clothing
* Basic Utiities
* Transport
* Other

*Errors :*

* All entries must be numeric
* none of the fields should be empty
* total of goals should be less than the budget

If the total of goals is not exactly equal to budget, but less than it, the user is prompted to either change goals or the extra amount would be added to savings goal.

Daily Entry

Once you click done on this screen, the amount you enter is subtracted from your goal, for the main screen displays. There is a checkbox on this screen for whether or not you have payed the tuition and living expenses.

*Required Entry Fields :*

* Food
* Entertainment
* Clothing
* Basic Utilities
* Transport
* Other

*Errors :*

* All entries must be numeric
* none of the fields should be empty

If the total of goals is not exactly equal to budget, but less than it, the user is prompted to either change goals or the extra amount would be added to savings goal.

Homepage

This page displays how much amount has been spent on each category, and how much is left out of you goal.

*Required Entry Fields :*

* Monthly Budget
* Tuition & Living Expense

[Budget goals]

* Food
* Entertainment
* Clothing
* Basic Utiities
* Transport
* Other

*Errors :*

* All entries must be numeric
* none of the fields should be empty
* total of goals should be less than the budget

If the total of goals is not exactly equal to budget, but less than it, the user is prompted to either change goals or the extra amount would be added to savings goal.

|  |
| --- |
| Source Code |
| from tkinter import \*  from tkinter.font import Font  from tkinter import messagebox  from sqlite3 import \*  import datetime  import calendar  from matplotlib import pyplot as plt  from matplotlib.figure import Figure  from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg, \  NavigationToolbar2Tk  from matplotlib import patches  #colour codes  blue = "#264653"  green = "#0D5E53"  yellow = "#F2C337"  orange = "#F49303"  red = "#FF4C3F"  peach="#F7AC94"  grey = "#BCE0CC"  red = "#FF4C3F"  pink = "#F29696"  #Current year and month  current\_year=int(datetime.date.today().strftime("%Y"))  current\_month=datetime.date.today().strftime("%B")  current\_month\_no = int(datetime.date.today().strftime("%m"))  #Initializing the main screen  def build\_screen():  global root,frame1  root = Tk()  root.title("d r á k o n : Your personal budget tracker")  root.attributes('-fullscreen',True)  root.bind('<Escape>',small\_screen)  canvas = Canvas(root,height=710,width=1250)  canvas.pack()  frame1 = Frame(root, bg = blue)  frame1.place(relwidth = 1, relheight = 1)  #def function for esc to small screen  def small\_screen(x) :  root.attributes('-fullscreen',False)  build\_screen()  #Fonts to be used  r18 = Font(family = "Bahnschrift", size = 18)  c22i = Font(family = "Courier New", size = 22 , slant = "italic")  c30 = Font(family = "Courier New", size = 30)  title = Font(family = "Rockwell", size = 100)  b20 = Font(family = "Bahnschrift", size = 20)  b16 = Font(family = "Bahnschrift", size = 16)  r24 = Font(family = "Rockwell", size = 24)  r20 = Font(family = "Rockwell", size = 20, slant = "italic")  r55 = Font(family = "Rockwell", size = 55)  b22 = Font(family = "Bahnschrift", size = 22)  g28 = Font(family = "Yu Gothic UI Semilight", size = 28)  g16b = Font(family = "Yu Gothic UI Semilight", size = 16, weight = "bold")  g45 = Font(family = "Yu Gothic UI Semilight", size = 45)  c60 = Font(family = "Courier New", size = 60)  c60i = Font(family = "Courier New", size = 60, slant = "italic")  r40 = Font(family = "Rockwell", size = 55)  r30 = Font(family = "Rockwell", size = 30)  b40 = Font(family = "Bahnschrift", size = 40)  g24 = Font(family = "Yu Gothic UI Semilight", size = 24)  c35 = Font(family = "Courier New", size = 35, weight = "bold")  c12 = Font(family = "Yu Gothic UI Semilight", size = 12)  def clear\_frame():  for widget in frame1.winfo\_children():  widget.destroy()  def textbox(frame,width=18,font=b20):  box = Entry(frame, selectborderwidth = "2px", bd = "1px", width = width, \  relief = SUNKEN, font = font, fg = blue)  return box  def asterisk\_textbox(frame,width=18,font=b20):  box = Entry(frame, selectborderwidth = "2px", bd = "1px", width = width, \  relief = SUNKEN, font = font, \  fg = blue, show = "\*")  return box    def login\_screen():    #Assigning function to the login button  def login\_function():  username=username\_entry.get()  password=password\_entry.get()  error=Label(login\_frame, font = g16b ,\  bg = green , fg = '#800000')  error.place(relx=0.5,rely=0.591,relwidth=1,anchor=CENTER)  cur.execute('SELECT Username,Password FROM userinfo WHERE Username==?'\  ,(username,))    if username=='' or password=='':  error.config(text = 'Invalid Input')  username\_entry.delete(0,END)  password\_entry.delete(0,END)  elif cur.fetchone()!=(username,password):  error.config(text = 'Incorrect username or password')  username\_entry.delete(0,END)  password\_entry.delete(0,END)  else:  main\_screen(username)  clear\_frame()  #logo  canvas = Canvas(frame1, width = 120, height = 120)  canvas.config(bg = blue, relief = FLAT, bd=0, highlightthickness=0)  canvas.place(relx = 0.33, rely = 0.38, anchor = CENTER)  picture = PhotoImage(file = "C:\\Users\\Dell\\Desktop\\project1\\blue.png")  logo = canvas.create\_image(60, 60, image = picture)    #app name  app\_name = Label(frame1, text="d r á k o n", \  font = title, \  bg = blue, foreground = yellow)  app\_name.place(relx = 0.33, rely = 0.55, anchor = CENTER)  app\_ = Label(frame1, text="personal budget tracker", \  font = c22i, \  bg = blue, foreground = yellow)  app\_.place(relx = 0.33, rely = 0.68 , anchor = CENTER)  #login/register frame  login\_frame = Frame(frame1, \  width = "9cm" , height = "10.5cm", \  bg = green, bd = "0px")  login\_frame.place(relx = 0.79, rely = 0.5, anchor = CENTER)    #username\_rname entry for logging in  username\_label = Label(login\_frame, text="Username", \  font = r18, \  bg = green, foreground = yellow)  username\_label.place(relx = 0.135, rely = 0.16, anchor = W)  username\_entry = textbox(login\_frame,18,r18)  username\_entry.place(relx = 0.5, rely = 0.25, anchor = CENTER)    #password entry for logging in  password\_label = Label(login\_frame, text="Password", \  font = r18, \  bg = green, foreground = yellow)  password\_label.place(relx = 0.135, rely = 0.39, anchor = W)  password\_entry = asterisk\_textbox(login\_frame,18,r18)  password\_entry.place(relx = 0.5, rely = 0.48, anchor = CENTER)  #login button  login = Button(login\_frame, text = " LOGIN ", \  bd = "0px", relief = FLAT, font = r18,\  bg = green, foreground = yellow, \  activebackground = green, activeforeground = peach,\  command=login\_function)  login.place(relx = 0.5, rely = 0.69, anchor = CENTER)  #register button  register = Button(login\_frame, text = " REGISTER ", \  bd = "0px", relief = FLAT, font = r18,\  bg = yellow, foreground = green, \  activebackground = yellow, activeforeground = blue,\  command = register\_screen)  register.place(relx = 0.5, rely = 0.84, anchor = CENTER)    root.mainloop()  def create\_table():  global cur,con  con=connect('mydatabase.db')  cur=con.cursor()  cur.execute('''CREATE TABLE IF NOT EXISTS userinfo(Username TEXT  PRIMARY KEY,Password TEXT,First\_name TEXT,Last\_name TEXT)''')  cur.execute('''CREATE TABLE IF NOT EXISTS budget\_info(Username TEXT,  Year INTEGER, Month TEXT, Monthly\_budget INTEGER,  Tution\_and\_living\_expenses INTEGER,Food\_budget INTEGER,  Entertainment\_budget INTEGER, Clothing\_budget INTEGER,  Basic\_utilities\_budget INTEGER,  Transport\_budget INTEGER, Other\_budget INTEGER,  Tution\_and\_living\_left INTEGER, Food\_left INTEGER,  Entertainment\_left INTEGER, Clothing\_left INTEGER,  Basic\_utilities\_left INTEGER, Transport\_left INTEGER,  Other\_left INTEGER,  Spent\_on\_Tution\_and\_living INTEGER,Spent\_on\_Food INTEGER,  Spent\_on\_Entertainment INTEGER,Spent\_on\_Clothing INTEGER,  Spent\_on\_Basic\_utilities INTEGER,  Spent\_on\_Transport INTEGER,Spent\_on\_Other INTEGER)''')  con.commit()  def register\_screen():    #Assigning function to done button  def done\_function():  username=username\_entry.get()  first\_name=first\_name\_entry.get()  last\_name=last\_name\_entry.get()  password1=password1\_entry.get()  password2=password2\_entry.get()  error = Label(register\_frame, text="",font = b20,bg = green, fg = '#800000')  error.place(relx = 0.5, rely = 0.804, relwidth=0.8, anchor = CENTER)    cur.execute("SELECT Username FROM userinfo WHERE Username=?",(username,))    if first\_name == '' or last\_name == '' or username == '' or\  password1 == '' or password2 == '' :  error.config(text="ERROR : Some required areas are empty")  elif cur.fetchone()!=None:  error.config(text="ERROR : Username not available")  elif password1!=password2 :  error.config(text="ERROR : Passwords do not match")  elif len(password1)<8:  error.config(text="ERROR : Password less than 8 characters")  else:  save\_userinfo(username,first\_name,last\_name,password1)  budget\_plan\_screen(username,first\_name)  #Function to save userinfo  def save\_userinfo(username,first\_name,last\_name,password):  cur.execute(" INSERT INTO userinfo VALUES (?,?,?,?)",( username ,\  password , first\_name , last\_name ))  month\_no=current\_month\_no  for year in range(current\_year,current\_year+10):  for month in range(month\_no,13):  cur.execute("INSERT INTO budget\_info VALUES(?,?,?,?,?,?,?,?,\  ?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)",(username,year,\  calendar.month\_name[month],0,0,0,0,0,0,0,0,0,0,0,0,\  0,0,0,0,0,0,0,0,0,0))  if month == 12:  month\_no = 1  con.commit()  def label\_func(text,rely):  label = Label(register\_frame, text=text, \  font = b20, bg = green, foreground = grey)  label.place(relx = 0.25, rely = rely, anchor = CENTER)  #frame  clear\_frame()  register\_frame = Frame(frame1, \  width = "19cm" , height = "17cm", bg = green, bd = "0px")  register\_frame.place(relx = 0.5, rely = 0.5, anchor = CENTER)  #labels  heading\_label = Label(register\_frame, text="Create your account", \  font = r24, bg = green, foreground = peach)  heading\_label.place(relx = 0.5, rely = 0.1, anchor = CENTER)  #First name  label\_func("First Name",0.22)  #Last name  label\_func("Last Name",0.345)  #Username  label\_func("Username",0.47)  #Password  label\_func("Password",0.595)  #Re enter password  label\_func("Re-Enter Password",0.72)  first\_name\_entry =textbox(register\_frame)  first\_name\_entry.place(relx = 0.45, rely = 0.22, anchor = W)  last\_name\_entry = textbox(register\_frame)  last\_name\_entry.place(relx = 0.45, rely = 0.345, anchor = W)  username\_entry = textbox(register\_frame)  username\_entry.place(relx = 0.45, rely = 0.47, anchor = W)  password1\_entry = asterisk\_textbox(register\_frame)  password1\_entry.place(relx = 0.45, rely = 0.595, anchor = W)  password2\_entry = asterisk\_textbox(register\_frame)  password2\_entry.place(relx = 0.45, rely = 0.72, anchor = W)  #Back button  back = Button(frame1, text = " BACK ", \  bd = "0px", relief = FLAT, font = b20,\  bg = peach, foreground = blue, \  activebackground = peach, activeforeground = green, \  command = login\_screen)  back.place(x = 30, y = 30, anchor = NW)  #done button  done = Button(register\_frame, text = " DONE ", \  bd = "0px", relief = FLAT, font = b20,\  bg = peach, foreground = green, \  activebackground = peach, activeforeground = blue,\  command = done\_function)  done.place(relx = 0.5, rely = 0.9, anchor = CENTER)  root.mainloop()  def budget\_plan\_screen(username,first):  def error(text):  Label(frame2, text = text, font = r20, \  bg = green, fg = '#800000').place(relx = 0.5, rely = 0.85,\  anchor = CENTER)  #Assigning function to done button  def done(budget,charges,food,entertainment,clothing,basic\_utilities,\  transport,other):    if budget=='' or charges=='' or \  food=='' or entertainment=='' or clothing=='' or \  basic\_utilities=='' or transport=='' or other=='':  error("Error : Required field(s) empty")  elif not((budget+charges+food+entertainment+clothing+basic\_utilities+\  transport+other).isdigit()):  error("Error : Invalid input")  elif int(charges) + int(food) + int(entertainment) + int(clothing) \  + int(basic\_utilities) + int(transport) + int(other) > int(budget) :  error("Error : Goals exceed budget")  else:  if int(charges) + int(food) + int(entertainment) + int(clothing)\  + int(basic\_utilities) + int(transport) + int(other) < int(budget) :  option=warning()  if option=="yes":  save\_budget(budget,charges,food,entertainment,clothing,\  basic\_utilities,transport,other)  main\_screen(username)  else:  save\_budget(budget,charges,food,entertainment,clothing,\  basic\_utilities,transport,other)  main\_screen(username)    def save\_budget(budget,charges,food,entertainment,clothing,basic\_utilities,\  transport,other):    cur.execute("""UPDATE budget\_info SET (Monthly\_budget,  Tution\_and\_living\_expenses, Food\_budget,  Entertainment\_budget, Clothing\_budget,  Basic\_utilities\_budget, Transport\_budget,  Other\_budget)=(?,?,?,?,?,?,?,?) WHERE Username=?""", \  (budget,charges,food,entertainment,clothing,basic\_utilities,\  transport,other,username))    cur.execute("""UPDATE budget\_info SET  (Tution\_and\_living\_left, Food\_left, Entertainment\_left,  Clothing\_left, Basic\_utilities\_left, Transport\_left,  Other\_left) = (?,?,?,?,?,?,?) WHERE Username=?""", \  (charges, food, entertainment, clothing, \  basic\_utilities, transport, other, username))    con.commit()    def warning():  warning = messagebox.askquestion("Unused amount", "You have an \  amount left unused.\nDo you want to add it to savings?")  return warning    def label\_func(text, rely, relx = 0.33, fg = grey):  label = Label(frame2, text = text, font = b20, bg = green, fg = fg)  label.place(relx = relx, rely = rely, anchor = CENTER)    #Initializing the screen  clear\_frame()  frame2 = Frame(frame1, bg = green, width = "22cm", height = "15.5cm")  frame2.place(relx = 0.5, rely = 0.56, anchor = CENTER)  #labels and textboxes  #WElcome label  welcome\_label = Label(frame1,text='Welcome {},'.format(first), \  bg = blue, fg = peach, font = r55)  welcome\_label.place(relx = 0.07, rely = 0.04)  #Budget  label\_func("Set your monthly budget : ", 0.075, 0.35)  budget\_entry = textbox(frame2,15)  budget\_entry.place(relx = 0.7, rely = 0.075, anchor = CENTER)  #Tution and living  label\_func("Tution and Living Expense : ", 0.165, 0.35)  charges\_entry = textbox(frame2,15)  charges\_entry.place(relx = 0.7, rely = 0.165, anchor = CENTER)  #Main label  label\_func("How much would you like to spend on", 0.245, 0.5, peach)  #Food  label\_func("Food",0.33)  food\_entry = textbox(frame2,15)  food\_entry.place(relx = 0.63,rely = 0.33, anchor = CENTER)  #Entertainment  label\_func("Entertainment",0.42)  entertainment\_entry = textbox(frame2,15)  entertainment\_entry.place(relx = 0.63, rely = 0.42, anchor = CENTER)  #Clothing  label\_func("Clothing",0.51)  clothing\_entry = textbox(frame2,15)  clothing\_entry.place(relx = 0.63, rely = 0.51, anchor = CENTER)  #Basic utilities  label\_func("Basic Utilities",0.6)  basic\_utilities\_entry= textbox(frame2,15)  basic\_utilities\_entry.place(relx=0.63, rely = 0.6, anchor = CENTER)  #Transport  label\_func("Transport",0.69)  transport\_entry = textbox(frame2,15)  transport\_entry.place(relx = 0.63, rely = 0.69, anchor = CENTER)  #Other  label\_func("Other",0.78)  other\_entry = textbox(frame2,15)  other\_entry.place(relx = 0.63, rely = 0.78, anchor = CENTER)  done\_button = Button(frame2, text = "DONE", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = green, activebackground = peach,\  activeforeground = blue,  command=lambda:done(budget\_entry.get(),charges\_entry.get(),\  food\_entry.get(), entertainment\_entry.get(),\  clothing\_entry.get(),\  basic\_utilities\_entry.get(),\  transport\_entry.get(), other\_entry.get()))  done\_button.place(relx = 0.5, rely=0.925, anchor = CENTER)  root.mainloop()  def daily\_entry\_screen(username):  def save\_daily\_entry(state,food,entertainment,clothing,basic\_utilities, \  transport,other):    cur.execute("""SELECT Tution\_and\_living\_expenses, Spent\_on\_Food,  Spent\_on\_Entertainment, Spent\_on\_Clothing,  Spent\_on\_Basic\_utilities, Spent\_on\_Transport,  Spent\_on\_Other FROM budget\_info WHERE USERNAME = ? AND  Year = ? AND Month = ?""",(username,current\_year,current\_month))  spent = cur.fetchone()  fee,food1,entertainment1,clothing1,basic\_utilities1,transport1,\  other1=spent[0],spent[1], spent[2],spent[3],spent[4],spent[5],spent[6]  state = state.get()  if state == 0:  current\_fee , spent\_on\_fee = fee , 0  elif state == 1:  current\_fee , spent\_on\_fee = 0 , fee    cur.execute("""UPDATE budget\_info SET ( Spent\_on\_Tution\_and\_living,  Spent\_on\_Food, Spent\_on\_Entertainment, Spent\_on\_Clothing,  Spent\_on\_Basic\_utilities, Spent\_on\_Transport,  Spent\_on\_Other ) = (?,?,?,?,?,?,?) WHERE Username=? AND  Year=? AND Month=?""",(spent\_on\_fee,food1+int(food), \  entertainment1+int(entertainment),clothing1+int(clothing),\  basic\_utilities1+int(basic\_utilities), \  transport1+int(transport),other1+int(other),username, \  current\_year, current\_month))  cur.execute("""SELECT Spent\_on\_Food, Spent\_on\_Entertainment,  Spent\_on\_Clothing, Spent\_on\_Basic\_utilities,Spent\_on\_Transport,  Spent\_on\_Other FROM budget\_info WHERE USERNAME = ? AND  Year = ? AND Month = ?""",(username,current\_year,current\_month))  spent = cur.fetchone()  food1,entertainment1,clothing1,basic\_utilities1,transport1,\  other1=spent[0],spent[1], spent[2], spent[3], spent[4], spent[5]  cur.execute("""SELECT Food\_budget, Entertainment\_budget, Clothing\_budget,  Basic\_utilities\_budget, Transport\_budget, Other\_budget FROM  budget\_info WHERE USERNAME=? AND Year=? AND Month = ?""", \  (username, current\_year, current\_month))  total=cur.fetchone()  food2,entertainment2,clothing2,basic\_utilities2,transport2,\  other2=total[0],total[1], total[2], total[3], total[4], total[5]  cur.execute("""UPDATE budget\_info SET (Tution\_and\_living\_left,Food\_left,  Entertainment\_left,Clothing\_left, Basic\_utilities\_left,  Transport\_left, Other\_left ) = (?,?,?,?,?,?,?) WHERE  Username = ? AND Year = ? AND Month = ?""", \  (current\_fee, food2-food1, entertainment2-entertainment1, \  clothing2-clothing1, basic\_utilities2-basic\_utilities1, \  transport2-transport1, other2-other1, username, \  current\_year, current\_month))  con.commit()  def done(state,food,entertainment,clothing,basic\_utilities,transport,other):  error = Label(frame2, text = "", font = r20, \  bg = green,fg = '#800000')  error.place(relx = 0.5, rely = 0.82, \  anchor = CENTER)    if food == '' or entertainment == '' or clothing == '' or \  basic\_utilities == '' or transport == '' or other == '':  error.config(text = "Error : Required field(s) empty")  elif not((food + entertainment + clothing + basic\_utilities + transport +\  other).isdigit()):  error.config(text = "Error : Invalid Input")  else:  save\_daily\_entry(state, food, entertainment, clothing, \  basic\_utilities, transport, other)  main\_screen(username)    def label\_func(text, rely, relx = 0.33, fg = grey):  label = Label(frame2,text = text, font = b20, bg = green, fg = fg)  label.place(relx = relx, rely = rely, anchor = CENTER)  #Initializing the screen  clear\_frame()  frame2 = Frame(frame1, bg = green, width = "22cm", height = "15.5cm")  frame2.place(relx = 0.5, rely = 0.5, anchor = CENTER)  #labels and textboxes  #Main label  label\_func("How much have you spent on", 0.105, 0.5, peach)  #Food  label\_func("Food", 0.21)  food\_entry = textbox(frame2,15)  food\_entry.place(relx = 0.63,rely = 0.21, anchor = CENTER)  #Entertainment  label\_func("Entertainment",0.315)  entertainment\_entry = textbox(frame2,15)  entertainment\_entry.place(relx = 0.63, rely = 0.315, anchor = CENTER)  #Clothing  label\_func("Clothing", 0.42)  clothing\_entry = textbox(frame2,15)  clothing\_entry.place(relx = 0.63, rely = 0.42, anchor = CENTER)  #Basic utilities  label\_func("Basic Utilities", 0.525)  basic\_utilities\_entry = textbox(frame2,15)  basic\_utilities\_entry.place(relx = 0.63, rely = 0.525, anchor = CENTER)  #Transport  label\_func("Transport", 0.63)  transport\_entry = textbox(frame2,15)  transport\_entry.place(relx = 0.63, rely = 0.63, anchor = CENTER)  #Other  label\_func("Other", 0.735)  other\_entry = textbox(frame2,15)  other\_entry.place(relx = 0.63, rely = 0.735, anchor = CENTER)  #Back button  back\_button = Button(frame1, text = "BACK", border = "0px", \  relief = FLAT, font = b20, bg = peach, fg = green, \  activebackground = peach, activeforeground = blue, \  command = lambda: main\_screen(username))  back\_button.place(x = 30, y = 30, anchor = NW )  state=IntVar()  charges\_checkbox=Checkbutton(frame2,text="Tution fee and hostel charges\  paid" , font = ("times new roman",16) , bg = green , variable = state, \  onvalue = 1, offvalue = 0 )  charges\_checkbox.place( relx = 0.75, rely = 0.9 , anchor = CENTER)  charges\_checkbox.select()  #Done button  done\_button = Button(frame2, text = "DONE", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = green, activebackground = peach,\  activeforeground = blue, \  command = lambda: done( state,food\_entry.get(), \  entertainment\_entry.get(), clothing\_entry.get(), \  basic\_utilities\_entry.get(), transport\_entry.get(), \  other\_entry.get()))  done\_button.place(relx = 0.5, rely=0.91, anchor = CENTER, relheight = 0.078)    root.mainloop()  def main\_screen(username):  global current\_year, current\_month, current\_month\_no  current\_year=int(datetime.date.today().strftime("%Y"))  current\_month=datetime.date.today().strftime("%B")  current\_month\_no = int(datetime.date.today().strftime("%m"))  click\_counter=0  #settings frame  def settings\_frame():  nonlocal click\_counter,settings\_frame  click\_counter+=1  if click\_counter%2 != 0:  settings\_frame = Frame(frame1,bg=yellow,width="7cm",height="5cm")  settings\_frame.place(relx = 0.975, rely = 0.1, anchor = NE)  change\_budget = Button(settings\_frame, text = "Change Budget", \  relief = FLAT, bd = "0px", font = b22, \  bg = yellow, activebackground = yellow, \  fg = blue, activeforeground = green, \  command = lambda: change\_budget\_screen(username))  change\_budget.place(relx = 0.5, rely = 0.2, anchor = CENTER)  change\_password = Button(settings\_frame, text = "Change Password", \  relief = FLAT, bd = "0px", font = b22, \  bg = yellow, activebackground = yellow, \  fg = blue, activeforeground = green,  command = lambda: change\_password\_screen(username))  change\_password.place(relx = 0.5, rely = 0.5, anchor = CENTER)  logout = Button(settings\_frame, text = "Logout", \  relief = FLAT, bd = "0px", font = b22, \  bg = yellow, activebackground = yellow, \  fg = blue, activeforeground = green, \  command = login\_screen)  logout.place(relx = 0.5, rely = 0.8, anchor = CENTER)    elif click\_counter%2==0:  settings\_frame.destroy()  def load\_data():  cur.execute("""SELECT Spent\_on\_Tution\_and\_living, Spent\_on\_food,  Spent\_on\_Entertainment, Spent\_on\_Clothing ,  Spent\_on\_Basic\_utilities, Spent\_on\_Transport, Spent\_on\_Other,  Tution\_and\_living\_left, Food\_left,  Entertainment\_left, Clothing\_left, Basic\_utilities\_left,  Transport\_left, Other\_left FROM budget\_info WHERE Username = ? \  AND Year = ? AND Month = ?""",(username, current\_year, \  current\_month))  info = cur.fetchone()  return info  def error(text,rely):  error=Label(frame1, text=text,font = b20,bg = blue, fg = red)  error.place(relx = 0.5, rely = rely,relwidth=0.8,anchor = CENTER)  def tab(text, relx, command):  tab = Button(frame2, text = text, \  border = "0px",height = "1", width = "15", \  relief = FLAT, font = b22, bg = green, \  activebackground = green, fg = peach, \  activeforeground = grey, \  command = command)  tab.place(anchor = CENTER, relx = relx, rely = 0.5)  def label(text):  label = Label(frame1,text = text, font = g28, bg = blue, \  fg = grey)  return label  #initializing the screen  clear\_frame()  frame2 = Frame(frame1, bg = green)  frame2.place(relwidth = 1,relheight = 0.1,relx = 0.5,rely = 0,anchor = N)    #screens buttons  #Budget Goals  tab("Daily Entry", 0.1, lambda: daily\_entry\_screen(username))  #month pie charts  tab("Monthly Review", 0.26, lambda: year\_selection(username))  #year review  tab("Yearly Analysis", 0.44, lambda: select\_year(username))  #ONSCREEN labels  date = Label(frame1, text=datetime.date.today().strftime("%d-%b-%Y"), \  font = b22, bg = blue, fg = grey)  date.place(relx = 0.015, rely = 0.11, anchor = NW)    spent\_label = label("Amount Spent")  spent\_label.place(relx = 0.49, rely = 0.175, anchor = CENTER)  left\_label = label("Amount Left")  left\_label.place(relx = 0.665, rely = 0.175, anchor = CENTER)  #load data  charges\_spent,food\_spent,entertainment\_spent,clothing\_spent, \  utilities\_spent, transport\_spent,other\_spent,chargesleft,foodleft, \  entertainmentleft, \  clothingleft,utilitiesleft,transportleft,otherleft =load\_data()  #Tution and living  charges = label("Tution and living")  charges.place(relx = 0.29, rely = 0.265, anchor = CENTER)  charges\_spent = label(charges\_spent)  charges\_spent.place(relx = 0.49, rely = 0.265, anchor = CENTER)  charges\_left = label(chargesleft)  charges\_left.place(relx = 0.665, rely = 0.265, anchor = CENTER)  #food  food = label("Food")  food.place(relx = 0.29, rely = 0.3675, anchor = CENTER)  food\_spent = label(food\_spent)  food\_spent.place(relx = 0.49, rely = 0.3675, anchor = CENTER)  food\_left = label(foodleft)  food\_left.place(relx = 0.665, rely = 0.3675, anchor = CENTER)  #entertainment  entertainment = label("Entertainment")  entertainment.place(relx = 0.29, rely = 0.47, anchor = CENTER)  entertainment\_spent = label(entertainment\_spent)  entertainment\_spent.place(relx = 0.49, rely = 0.47, anchor = CENTER)  entertainment\_left = label(entertainmentleft)  entertainment\_left.place(relx = 0.665, rely = 0.47, anchor = CENTER)  #clothing  clothing = label("Clothing")  clothing.place(relx = 0.29, rely = 0.5725, anchor = CENTER)  clothing\_spent = label(clothing\_spent)  clothing\_spent.place(relx = 0.49, rely = 0.5725, anchor = CENTER)  clothing\_left = label(clothingleft)  clothing\_left.place(relx = 0.665, rely = 0.5725, anchor = CENTER)  #utilities  utilities = label("Basic Utilities")  utilities.place(relx = 0.29, rely = 0.675, anchor = CENTER)  utilities\_spent = label(utilities\_spent)  utilities\_spent.place(relx = 0.49, rely = 0.675, anchor = CENTER)  utilities\_left = label(utilitiesleft)  utilities\_left.place(relx = 0.665, rely = 0.675, anchor = CENTER)  #transport  transport = label("Transport")  transport.place(relx = 0.29, rely = 0.7775, anchor = CENTER)  transport\_spent = label(transport\_spent)  transport\_spent.place(relx = 0.49, rely = 0.7775, anchor = CENTER)  transport\_left = label(transportleft)  transport\_left.place(relx = 0.665, rely = 0.7775, anchor = CENTER)  #other  other = label("Other")  other.place(relx = 0.29, rely = 0.88, anchor = CENTER)  other\_spent = label(other\_spent)  other\_spent.place(relx = 0.49, rely = 0.88, anchor = CENTER)  other\_left = label(otherleft)  other\_left.place(relx = 0.665, rely = 0.88, anchor = CENTER)  #Errors  if foodleft<0:  error("Food budget has been exceeded by "+ str(abs(foodleft)),0.41875)  food\_left.config(text=0)  if entertainmentleft<0:  error("Entertainment budget has been exceeded by "+ \  str(abs(entertainmentleft)),0.52125)  entertainment\_left.config(text=0)  if clothingleft<0:  error("Clothing budget has been exceeded by " + \  str(abs(clothingleft)),0.62375)  clothing\_left.config(text=0)  if utilitiesleft<0:  error("Basic utilities budget has been exceeded by " + \  str(abs(utilitiesleft)),0.72625)  utilities\_left.config(text=0)  if transportleft<0:  error("Transport budget has been exceeded by " + \  str(abs(transportleft)),0.82875)  transport\_left.config(text=0)  if otherleft<0:  error("Other budget has been exceeded by " + \  str(abs(otherleft)),0.93125)  other\_left.config(text=0)  if foodleft + entertainmentleft + clothingleft + utilitiesleft + \  transportleft + otherleft + chargesleft < 0:  messagebox.showwarning("WARNING","You have exceeded your total budget.\n\  Any extra amount spent will be considered as debt.")  #settings button  photo = PhotoImage(file = "C:\\Users\\Dell\\Desktop\\project1\\button.png")  setting\_button = Button(frame2, image = photo, highlightthickness = 0, \  bd = "0px", relief = FLAT, command = settings\_frame)  setting\_button.place(anchor = CENTER, relx = 0.95, rely = 0.5)    root.mainloop()  def change\_password\_screen(username):  def done():  cur.execute("SELECT Password FROM userinfo WHERE Username=?", \  (username,))  password = cur.fetchone()  password = password[0]  old\_password, new\_password, re\_enter\_password = password\_textbox.get(),\  new\_password\_textbox.get(), re\_enter\_textbox.get()  error = Label(change\_password, text = "", font = r20, \  bg = green, fg = '#800000')  error.place(relx = 0.5, rely = 0.72, anchor = CENTER)  if old\_password == '' or new\_password == '' or re\_enter\_password =='' :  error.config(text = "Error : Required field(s) empty")  elif old\_password != password :  error.config(text = "Error : Incorrect password")  elif new\_password != re\_enter\_password:  error.config(text = "Error : Passwords do not match")  elif len(new\_password) < 8:  error.config(text = "Error : Password less than 8 characters")  else:  cur.execute("UPDATE userinfo SET Password = ? WHERE Username = ?", \  (new\_password, username))  main\_screen(username)  def inner\_label(text,rely):  label = Label(change\_password,text = text,\  font = b22, bg = green, fg = peach)  label.place(relx = 0.25, rely = rely, anchor = CENTER)    clear\_frame()  change\_password = Frame(frame1, bg = green)  change\_password.place(relwidth = 0.55, relheight = 0.5,\  relx = 0.5, rely = 0.6, anchor = CENTER)  #labels  username\_label = Label(frame1,text = "Username",\  font = c60, bg = blue, fg = yellow)  username\_label.place(relx = 0.5, rely = 0.15, anchor = CENTER)  #personal label would be the actual username of the person  personal\_label = Label(frame1,text = username,\  font = c60i, bg = blue, fg = yellow)  personal\_label.place(relx = 0.5, rely = 0.25, anchor = CENTER)  #inside change password frame  inner\_label("Enter Old Password", 0.15)  inner\_label("Enter New Password", 0.35)  inner\_label("Re-enter New Password", 0.55)  password\_textbox = asterisk\_textbox(change\_password,18,b22)  password\_textbox.place(relx = 0.75, rely = 0.15, anchor = CENTER)  new\_password\_textbox = asterisk\_textbox(change\_password,18,b22)  new\_password\_textbox.place(relx = 0.75, rely = 0.35, anchor = CENTER)  re\_enter\_textbox = asterisk\_textbox(change\_password,18,b22)  re\_enter\_textbox.place(relx = 0.75, rely = 0.55, anchor = CENTER)  #back button  back = Button(frame1, text = " BACK ", \  bd = "0px", relief = FLAT, font = b22,\  bg = peach, foreground = green, \  activebackground = peach, activeforeground = blue, \  command = lambda: main\_screen(username))  back.place(x = 30, y = 30, anchor = NW)  #done button  done = Button(change\_password, text = " DONE ", \  bd = "0px", relief = FLAT, font = b22,\  bg = peach, foreground = green, \  activebackground = peach, activeforeground = blue, \  command = done)  done.place(relx = 0.5, rely = 0.85, anchor = CENTER)  root.mainloop()  def change\_budget\_screen(username):  def error(text):  Label(frame2, text = text, font = r20, \  bg = green, fg = '#800000').place(relx = 0.5, rely = 0.85,\  anchor = CENTER)  #Assigning function to done button  def done(budget,charges,food,entertainment,clothing,basic\_utilities,\  transport,other):    if budget=='' or charges=='' or \  food=='' or entertainment=='' or clothing=='' or \  basic\_utilities=='' or transport=='' or other=='':  error("Error : Required field(s) empty")  elif not((budget+charges+food+entertainment+clothing+basic\_utilities+\  transport+other).isdigit()):  error("Error : Invalid input")  elif int(charges) + int(food) + int(entertainment) + int(clothing) \  + int(basic\_utilities) + int(transport) + int(other) > int(budget) :  error("Error : Goals exceed budget")  else:  if int(charges) + int(food) + int(entertainment) + int(clothing)\  + int(basic\_utilities) + int(transport) + int(other) < int(budget) :  option=warning()  if option=="yes":  save\_budget(budget,charges,food,entertainment,clothing,\  basic\_utilities,transport,other)  main\_screen(username)  else:  save\_budget(budget,charges,food,entertainment,clothing,\  basic\_utilities,transport,other)  main\_screen(username)    def save\_budget(budget,charges,food,entertainment,clothing,basic\_utilities,\  transport,other):  cur.execute("SELECT Year FROM budget\_info WHERE Username=?",(username,))  year\_list = cur.fetchall()  year\_limit = year\_list[-1][0]  month\_no=current\_month\_no  for year in range(current\_year,year\_limit+1):  for month in range(month\_no,13):  cur.execute("""UPDATE budget\_info SET (Monthly\_budget,  Tution\_and\_living\_expenses, Food\_budget,  Entertainment\_budget, Clothing\_budget,  Basic\_utilities\_budget, Transport\_budget,  Other\_budget)=(?,?,?,?,?,?,?,?) WHERE Username=?  AND Year=? AND Month=?""",(budget,charges,food, \  entertainment,clothing,\  basic\_utilities,transport,other,username,\  year,calendar.month\_name[month]))  cur.execute("""SELECT Spent\_on\_Tution\_and\_living, Spent\_on\_Food,  Spent\_on\_Entertainment, Spent\_on\_Clothing,  Spent\_on\_Basic\_utilities,Spent\_on\_Transport,  Spent\_on\_Other FROM budget\_info WHERE USERNAME = ?  AND Year = ? AND Month = ?""",(username, year, \  calendar.month\_name[month]))  spent = cur.fetchone()  fee\_spent, food\_spent, entertainment\_spent, clothing\_spent, \  basic\_utilities\_spent, transport\_spent, other\_spent = spent[0], \  spent[1], spent[2], spent[3], spent[4], spent[5], spent[6]  cur.execute("""UPDATE budget\_info SET (Tution\_and\_living\_left,  Food\_left, Entertainment\_left, Clothing\_left,  Basic\_utilities\_left, Transport\_left, Other\_left ) =  (?,?,?,?,?,?,?) WHERE Username = ? AND Year = ? AND \  Month = ?""", (int(charges)-fee\_spent, \  int(food)-food\_spent, \  int(entertainment)-entertainment\_spent, \  int(clothing)-clothing\_spent, int(basic\_utilities)-\  basic\_utilities\_spent, int(transport)-transport\_spent,\  int(other)-other\_spent, username, year, \  calendar.month\_name[month]))  if month == 12:  month\_no = 1  con.commit()  def warning():  warning = messagebox.askquestion("Unused Amount", "You have an \  amount left unused.\nDo you want to add it to savings?")  return warning  def label\_func(text, rely, relx = 0.33, fg = grey):  label = Label(frame2, text = text, font = b20, bg = green, fg = fg)  label.place(relx = relx, rely = rely, anchor = CENTER)  #Initializing the screen  clear\_frame()  frame2 = Frame(frame1, bg = green, width = "22cm", height = "15.5cm")  frame2.place(relx = 0.5, rely = 0.56, anchor = CENTER)  #labels and textboxes  #Budget  label\_func("Set your monthly budget : ", 0.075, 0.35)  budget\_entry = textbox(frame2,15)  budget\_entry.place(relx = 0.7, rely = 0.075, anchor = CENTER)  #Tution and living  label\_func("Tution and Living Expense : ", 0.165, 0.35)  charges\_entry = textbox(frame2,15)  charges\_entry.place(relx = 0.7, rely = 0.165, anchor = CENTER)  #Main label  label\_func("How much would you like to spend on", 0.245, 0.5, peach)  #Food  label\_func("Food",0.33)  food\_entry = textbox(frame2,15)  food\_entry.place(relx = 0.63,rely = 0.33, anchor = CENTER)  #Entertainment  label\_func("Entertainment",0.42)  entertainment\_entry = textbox(frame2,15)  entertainment\_entry.place(relx = 0.63, rely = 0.42, anchor = CENTER)  #Clothing  label\_func("Clothing",0.51)  clothing\_entry = textbox(frame2,15)  clothing\_entry.place(relx = 0.63, rely = 0.51, anchor = CENTER)  #Basic utilities  label\_func("Basic Utilities",0.6)  basic\_utilities\_entry= textbox(frame2,15)  basic\_utilities\_entry.place(relx=0.63, rely = 0.6, anchor = CENTER)  #Transport  label\_func("Transport",0.69)  transport\_entry = textbox(frame2,15)  transport\_entry.place(relx = 0.63, rely = 0.69, anchor = CENTER)  #Other  label\_func("Other",0.78)  other\_entry = textbox(frame2,15)  other\_entry.place(relx = 0.63, rely = 0.78, anchor = CENTER)  done\_button = Button(frame2, text = "DONE", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = green, activebackground = peach,\  activeforeground = blue,  command=lambda:done(budget\_entry.get(),charges\_entry.get(),\  food\_entry.get(), entertainment\_entry.get(),\  clothing\_entry.get(),\  basic\_utilities\_entry.get(),\  transport\_entry.get(), other\_entry.get()))  done\_button.place(relx = 0.5, rely=0.925, anchor = CENTER)    back = Button(frame1, text = " BACK ", \  bd = "0px", relief = FLAT, font = b20,\  bg = peach, foreground = green, \  activebackground = peach, activeforeground = blue, \  command = lambda: main\_screen(username))  back.place(x = 30, y = 30, anchor = NW)    root.mainloop()  def select\_category(username, year):  def category(text,rely,command):  button = Button(frame1, text = text, bd = "0px", \  font = r30, relief = FLAT, bg = blue, fg = peach, \  activebackground = blue, activeforeground = pink, \  command = command)  button.place(relx = 0.5, rely = rely, anchor = CENTER)  #label  clear\_frame()  select = Label(frame1, text = "Select a category", font = c30, \  bg = blue, fg = grey)  select.place(relx = 0.5, rely = 0.1, anchor = CENTER)  #Category buttons  category("Tution and Living", 0.2, \  lambda: graph(username,"Tution and Living", year))  category("Food", 0.3, lambda: graph(username,"Food", year))  category("Entertainment",0.4, lambda: graph(username,"Entertainment", \  year))  category("Clothing", 0.5, lambda: graph(username,"Clothing", year))  category("Basic Utilities", 0.6, \  lambda: graph(username,"Basic Utilities", \  year))  category("Transport", 0.7, lambda: graph(username, "Transport", year))  category("Other", 0.8, lambda: graph(username,"Other", year))  #Back button  back\_button = Button(frame1, text = "BACK", bd = "0px", \  relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = lambda: select\_year(username))  back\_button.place( x = 30, y = 30, anchor = NW)  root.mainloop()  def select\_year(username):  def select\_func():  if years\_listbox.curselection() != ():  year\_choice = years\_listbox.get(years\_listbox.curselection())  select\_category(username, year\_choice)  else:  Label(frame1, text = "Please click on the year to select it.", \  font = r20, bg = blue, fg = "#800000").place(relx = 0.5, \  rely = 0.635, anchor = CENTER)  #Initializing the screen  clear\_frame()  label = Label(frame1, text = "Select the year", font = c60, \  bg = blue, fg = grey)  label.place(rely = 0.4, relx = 0.5, anchor = CENTER)  #list of years  years\_listbox = Listbox(frame1, font = c60, bd = "0px", relief = FLAT, \  bg = blue, fg = yellow, justify = "center", \  selectmode = SINGLE, selectbackground = green)  years\_listbox.place(relx = 0.5, rely = 0.55, anchor = CENTER, \  relwidth = 0.17, relheight = 0.1)#Adding scroll bar  scrollbar = Scrollbar(years\_listbox, orient = "vertical", \  command = years\_listbox.yview)  scrollbar.pack(side=RIGHT,fill=Y)  years\_listbox.config(yscrollcommand = scrollbar.set)    #Adding options to list box  cur.execute("SELECT Year FROM budget\_info WHERE Username=?",(username,))  year\_list = cur.fetchall()  first\_year = year\_list[0][0]  for year in range(first\_year, current\_year+1):  years\_listbox.insert(END, year)  #Buttons  back\_button = Button(frame1, text = "BACK", bd = "0px", \  relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = lambda: main\_screen(username))  back\_button.place( x = 30, y = 30, anchor = NW)  select\_button = Button(frame1, text = "SELECT", bd = "0px", \  relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = select\_func)  select\_button.place(relx = 0.5, rely = 0.7, anchor = CENTER)  root.mainloop()  def graph(username, choice, year):  def back\_func():  bar.destroy()  canvas2.get\_tk\_widget().destroy()  select\_category(username, year)  #Initializing the screen  clear\_frame()  back\_button = Button(frame1, text = "BACK", bd = "0px", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = back\_func)  back\_button.place(x = 30, y = 30, anchor = NW)  #Category  if choice == 'Tution and Living':  cur.execute("""SELECT Tution\_and\_living\_expenses,  Spent\_on\_Tution\_and\_living FROM budget\_info WHERE  Username = ? AND Year = ?""",(username,year))  elif choice == 'Food':  cur.execute("""SELECT Food\_budget,Spent\_on\_Food FROM budget\_info WHERE  Username = ? AND Year = ?""",(username,year))  elif choice == 'Entertainment':  cur.execute("""SELECT Entertainment\_budget,Spent\_on\_Entertainment  FROM budget\_info WHERE  Username = ? AND Year = ?""",(username,year))  elif choice == 'Clothing':  cur.execute("""SELECT Clothing\_budget,Spent\_on\_Clothing FROM budget\_info  WHERE Username = ? AND Year = ?""",(username,year))  elif choice == 'Basic Utilities':  cur.execute("""SELECT Basic\_utilities\_budget,Spent\_on\_Basic\_utilities  FROM budget\_info WHERE  Username = ? AND Year = ?""",(username,year))  elif choice == 'Transport':  cur.execute("""SELECT Transport\_budget,  Spent\_on\_Transport FROM budget\_info  WHERE Username = ? AND Year = ?""",(username,year))  elif choice == 'Other':  cur.execute("""SELECT Other\_budget,Spent\_on\_Other FROM budget\_info WHERE  Username = ? AND Year = ?""",(username,year))  #Plotting the graph  y=cur.fetchall()  ln=len(y)  x=[calendar.month\_name[month] for month in range(13-ln,13)]  y1=[value[0] for value in y]  y2=[value[1] for value in y]  fig = Figure(figsize = (5,4),dpi = 100)  subplot=fig.add\_subplot(111)  subplot.plot(x,y1,marker='o',label = choice + " goal",color=pink)  subplot.plot(x,y2,marker='o',label = choice + " spent",color=yellow)  canvas2 = FigureCanvasTkAgg(fig, master =root)  canvas2.draw()  canvas2.get\_tk\_widget().place(relx=0.5,rely=0.5,relheight=0.8,relwidth=0.8,\  anchor =CENTER)  bar = NavigationToolbar2Tk(canvas2,root)  bar.update()    subplot.set\_title(year,fontsize=26,color=grey)  subplot.grid(True)  fig.legend()  fig.tight\_layout()  subplot.set\_facecolor(green)  fig.set\_facecolor(green)  root.mainloop()  def year\_selection(username):  def select\_func():  if years\_listbox.curselection() != ():  year\_choice = years\_listbox.get(years\_listbox.curselection())  month\_selection(username, year\_choice)  else:  Label(frame1, text = "Please click on the year to select it.", \  font = r20, bg = blue, fg = "#800000").place(relx = 0.5, \  rely = 0.635, anchor = CENTER)  #Initializing the screen  clear\_frame()  label = Label(frame1, text = "Select the year", font = c60, \  bg = blue, fg = grey)  label.place(rely = 0.4, relx = 0.5, anchor = CENTER)  #list of years  years\_listbox = Listbox(frame1, font = c60, bd = "0px", relief = FLAT, \  bg = blue, fg = yellow, justify = "center", \  selectmode = SINGLE, selectbackground = green)  years\_listbox.place(relx = 0.5, rely = 0.55, anchor = CENTER, \  relwidth = 0.17, relheight = 0.1)  #Adding scroll bar  scrollbar = Scrollbar(years\_listbox, orient = "vertical", \  command = years\_listbox.yview)  scrollbar.pack(side=RIGHT,fill=Y)  years\_listbox.config(yscrollcommand = scrollbar.set)    #Adding options to list box  cur.execute("SELECT Year FROM budget\_info WHERE Username=?",(username,))  year\_list = cur.fetchall()  first\_year = year\_list[0][0]  for year in range(first\_year, current\_year+1):  years\_listbox.insert(END, year)  #Buttons  back\_button = Button(frame1, text = "BACK", bd = "0px", \  relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = lambda: main\_screen(username))  back\_button.place( x = 30, y = 30, anchor = NW)  select\_button = Button(frame1, text = "SELECT", bd = "0px", \  relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = select\_func)  select\_button.place(relx = 0.5, rely = 0.7, anchor = CENTER)  root.mainloop()  def month\_selection(username, year):  def btn\_func(month):  cur.execute("""SELECT \* FROM budget\_info WHERE Username == ? AND  Year == ? AND Month ==?""",(username, year, month))  if cur.fetchone() == None:  messagebox.showerror("Error","No data to display")  else:  pie\_chart(username, year, month)  def button(month):  button = Button(frame1, text = month, bd = "0px", \  font = r30, relief = FLAT, bg = blue, fg = peach,\  activebackground = blue, activeforeground = pink, \  command = lambda: btn\_func(month))  return button  #Initializing the screen  clear\_frame()  #back button  back\_button = Button(frame1, text = "BACK", bd = "0px", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = lambda: year\_selection(username))  back\_button.place( x = 30, y = 30, anchor = NW)  #label  select = Label(frame1, text = "Select a month", font = c30, bg = blue, \  fg = grey)  select.place(relx = 0.5, rely = 0.15, anchor = CENTER)  #month buttons  jan = button("January")  jan.place(relx = 0.35, rely = 0.25, anchor = CENTER)  feb = button("February")  feb.place(relx = 0.35, rely = 0.36, anchor = CENTER)  mar = button("March")  mar.place(relx = 0.35, rely = 0.47, anchor = CENTER)  apr = button("April")  apr.place(relx = 0.35, rely = 0.58, anchor = CENTER)  may = button("May")  may.place(relx = 0.35, rely = 0.69, anchor = CENTER)  jun = button("June")  jun.place(relx = 0.35, rely = 0.8, anchor = CENTER)  jul = button("July")  jul.place(relx = 0.65, rely = 0.25, anchor = CENTER)    aug = button("August")  aug.place(relx = 0.65, rely = 0.36, anchor = CENTER)    sep = button("September")  sep.place(relx = 0.65, rely = 0.47, anchor = CENTER)    octo = button("October")  octo.place(relx = 0.65, rely = 0.58, anchor = CENTER)  nov = button("November")  nov.place(relx = 0.65, rely = 0.69, anchor = CENTER)    dec = button("December")  dec.place(relx = 0.65, rely = 0.8, anchor = CENTER)    #running the proram  root.mainloop()  def pie\_chart(username, year, month):  def back\_func():  try:  debt\_label.destroy()  except:  None  legend\_frame.destroy()  canvas3.get\_tk\_widget().destroy()  month\_selection(username, year)  def legend(text,rely):  label = Label(legend\_frame, text = text, font = c12, \  bg = blue, fg = grey)  label.place(rely = rely, relx = 0.3, anchor = W)  def color(color,rely):  color = Frame(legend\_frame, bg = color, width = "1cm", height = "0.5cm")  color.place(relx = 0.16, rely = rely, anchor = CENTER)  #Initializing the screen  clear\_frame()    back\_button = Button(frame1, text = "BACK", bd = "0px", \  border = "0px", relief = FLAT, font = b20, \  bg = peach, fg = blue, \  activebackground = pink, activeforeground = blue, \  command = back\_func )  back\_button.place( x = 30, y = 30, anchor = NW)  cur.execute("""SELECT Monthly\_budget,Tution\_and\_living\_expenses,Food\_budget,  Entertainment\_budget, Clothing\_budget, Basic\_utilities\_budget,  Transport\_budget, Other\_budget, Spent\_on\_Tution\_and\_living,  Spent\_on\_Food, Spent\_on\_Entertainment, Spent\_on\_Clothing,  Spent\_on\_Basic\_utilities, Spent\_on\_Transport, Spent\_on\_Other  FROM budget\_info WHERE Username = ? AND Year = ? AND  Month = ?""",(username, year, month))    record = cur.fetchone()  total\_budget = record[0]  goals = [record[x] for x in range(1,8)]  colors1 = ["#0F52BA", "#5ac981", "#ce361e", "#f97a25", \  "#3597CC", "#037272", grey]  if sum(goals) < total\_budget:  savings = total\_budget - sum(goals)  goals.append(savings)  colors1.append("#efca08")    fig = Figure(figsize = (4,4),dpi = 100)  subplot=fig.add\_subplot(121)  subplot.pie(goals, startangle = 45, colors = colors1, \  textprops = {'color': 'w'})  circle1 = patches.Circle((0,0),0.6,color = blue)  subplot.add\_artist(circle1)  canvas3 = FigureCanvasTkAgg(fig, master =root)  canvas3.draw()  canvas3.get\_tk\_widget().place(relx=0.44, rely=0.55, relheight=0.8, \  relwidth=1,anchor =CENTER)  subplot.set\_title("Budget goals",fontsize=26,color=grey)  fig.tight\_layout()  subplot.set\_facecolor(blue)  fig.set\_facecolor(blue)  spent = [record[x] for x in range(8,15)]  colors2 = ["#0F52BA", "#5ac981", "#ce361e", "#f97a25", \  "#3597CC", "#037272", grey]  if sum(spent) < total\_budget:  savings = total\_budget - sum(spent)  spent.append(savings)  colors2.append("#efca08")  subplot1=fig.add\_subplot(1,2,2)  subplot1.pie(spent, startangle = 45, colors = colors2, \  textprops = {'color': 'w'})  circle2 = patches.Circle((0,0),0.6,color = blue)  subplot1.add\_artist(circle2)  subplot1.set\_title("Expenditure",fontsize=26,color=grey)  subplot1.set\_facecolor(blue)  #Onscreen headings  heading = Label(frame1, text = str(month) +' '+ str(year), fg = grey, \  bg =blue, font = b40)  heading.place(relx = 0.5, rely = 0.059, anchor = CENTER)  if sum(spent) > total\_budget:  debt = sum(spent) - total\_budget  debt\_label = Label(root, text = "Debt = "+str(debt), fg = grey, \  bg = blue, font = b20)  debt\_label.place(relx = 0.75, rely = 0.85, anchor = CENTER)  #Legend  legend\_frame = Frame(root, bg = blue, width = "5cm", height = "8cm")  legend\_frame.place(relx = 0.5, rely = 0.5, anchor = CENTER)  #Tution and living  legend("Tuition & Living", 0.1)  color("#0F52BA", 0.1)  #Food  legend("Food", 0.2)  color("#5AC981", 0.2)  #Entertainment  legend("Entertainment", 0.3)  color("#CE361E", 0.3)  #Clothing  legend("Clothing", 0.4)  color("#F97A25", 0.4)  #Basic Utilities  legend("Basic Utilities", 0.5)  color("#3597CC", 0.5)  #Transport  legend("Transport", 0.6)  color("#037272", 0.6)  #Other  legend("Other", 0.7)  color("#93C4A9", 0.7)  #Savings  legend("Savings", 0.8)  color("#EFCA08", 0.8)    root.mainloop()  create\_table()  login\_screen() |