DATABASE MANAGEMENT SYSTEMS

PROJECT REPORT

INVENTORY MANAGEMENT SYSTEM

Inventory Management

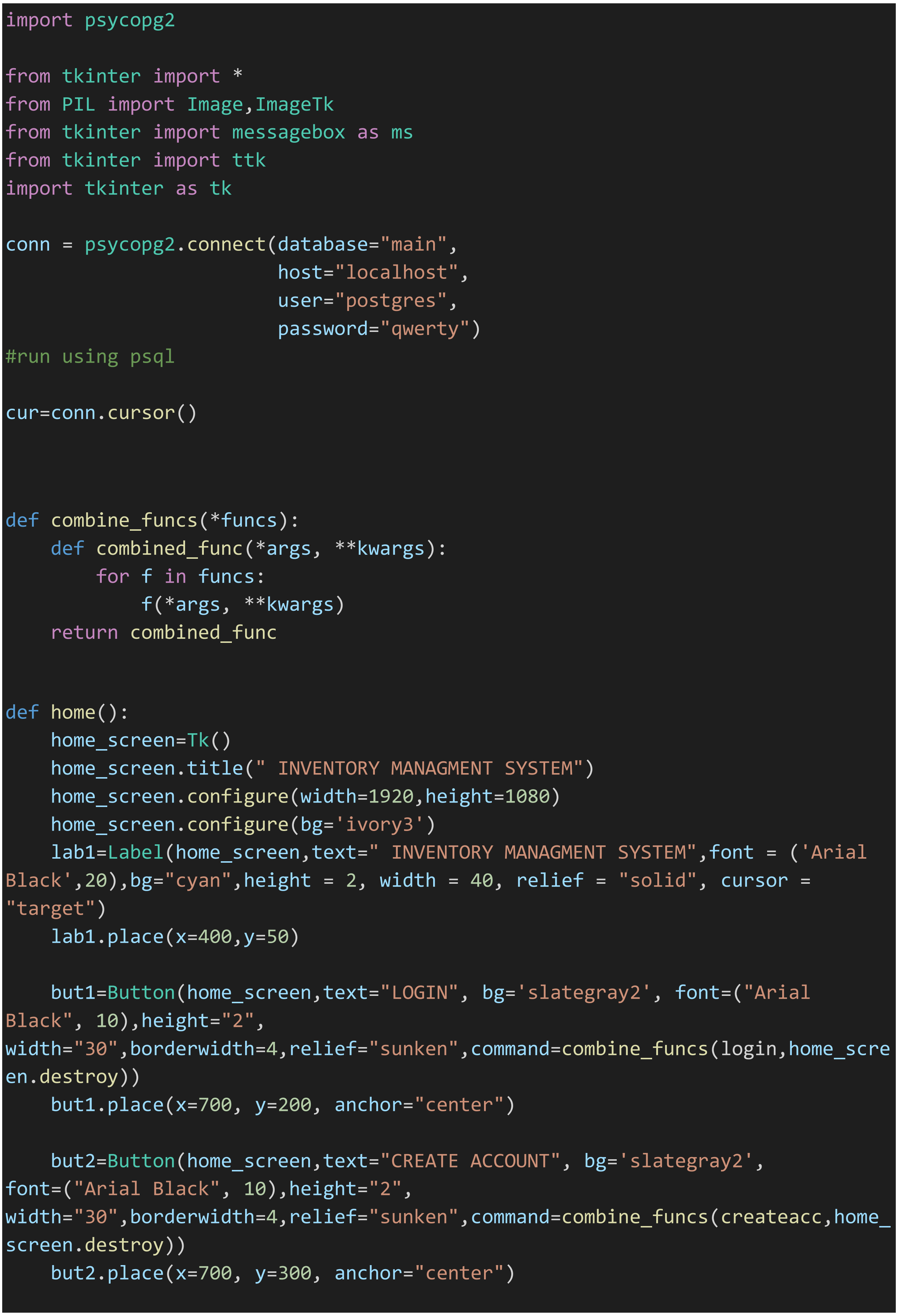
['in-vən-,tór-ē 'ma-nij-mənt]

The process of ordering, storing, using, and selling a company's raw materials, components, and finished products.

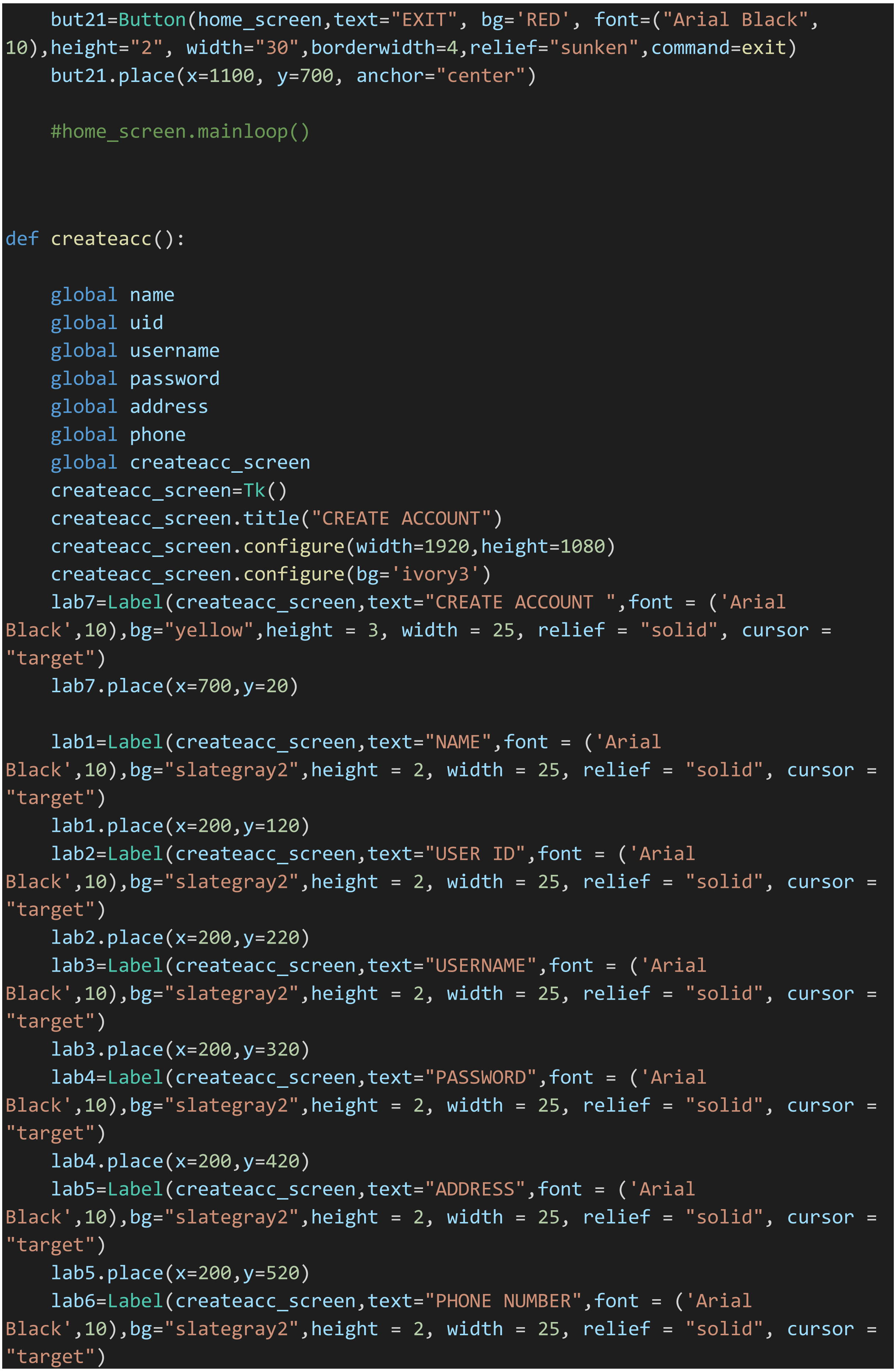
CHIRAG V

(PES1UG20EC252)

ABSTRACT:



```
import psycopg2

from tkinter import *
from PIL import Image,ImageTk
from tkinter import messagebox as ms
from tkinter import ttk
import tkinter as tk

conn = psycopg2.connect(database="main",
                        host="localhost",
                        user="postgres",
                        password="qwerty")
#run using psql

cur=conn.cursor()



def combine_funcs(*funcs):
    def combined_func(*args, **kwargs):
        for f in funcs:
            f(*args, **kwargs)
    return combined_func


def home():
    home_screen=Tk()
    home_screen.title(" INVENTORY MANAGMENT SYSTEM")
    home_screen.configure(width=1920,height=1080)
    home_screen.configure(bg='ivory3')
    lab1=Label(home_screen,text=" INVENTORY MANAGMENT SYSTEM",font = ('Arial Black',20),bg="cyan",height = 2, width = 40, relief = "solid", cursor = "target")
    lab1.place(x=400,y=50)

    but1=Button(home_screen,text="LOGIN", bg='slategray2', font=("Arial Black", 10),height="2", width="30",borderwidth=4,relief="sunken",command=combine_funcs(login,home_screen.destroy))
    but1.place(x=700, y=200, anchor="center")

    but2=Button(home_screen,text="CREATE ACCOUNT", bg='slategray2', font=("Arial Black", 10),height="2", width="30",borderwidth=4,relief="sunken",command=combine_funcs(createacc,home_screen.destroy))
    but2.place(x=700, y=300, anchor="center")
```



```
    but21=Button(home_screen,text="EXIT", bg='RED', font=("Arial Black",
10),height="2", width="30",borderwidth=4,relief="sunken",command=exit)
    but21.place(x=1100, y=700, anchor="center")

    #home_screen.mainloop()



def createacc():

    global name
    global uid
    global username
    global password
    global address
    global phone
    global createacc_screen
    createacc_screen=Tk()
    createacc_screen.title("CREATE ACCOUNT")
    createacc_screen.configure(width=1920,height=1080)
    createacc_screen.configure(bg='ivory3')
    lab7=Label(createacc_screen,text="CREATE ACCOUNT ",font = ('Arial
Black',10),bg="yellow",height = 3, width = 25, relief = "solid", cursor =
"target")
    lab7.place(x=700,y=20)

    lab1=Label(createacc_screen,text="NAME",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab1.place(x=200,y=120)
    lab2=Label(createacc_screen,text="USER ID",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab2.place(x=200,y=220)
    lab3=Label(createacc_screen,text="USERNAME",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab3.place(x=200,y=320)
    lab4=Label(createacc_screen,text="PASSWORD",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab4.place(x=200,y=420)
    lab5=Label(createacc_screen,text="ADDRESS",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab5.place(x=200,y=520)
    lab6=Label(createacc_screen,text="PHONE NUMBER",font = ('Arial
Black',10),bg="slategray2",height = 2, width = 25, relief = "solid", cursor =
"target")
```

```
    lab6.place(x=200,y=620)


    name_var=StringVar()
    uid_var=StringVar()
    username_var=StringVar()
    password_var=StringVar()
    address_var=StringVar()
    phone_var=StringVar()



    name = Entry(createacc_screen,textvariable = name_var,
font=('calibre',15,'normal'))
    name.place(x=600,y=120)


    uid = Entry(createacc_screen,textvariable = uid_var,
font=('calibre',15,'normal'))
    uid.place(x=600,y=220)

    username = Entry(createacc_screen,textvariable = username_var,
font=('calibre',15,'normal'))
    username.place(x=600,y=320)


    password = Entry(createacc_screen,textvariable = password_var,
font=('calibre',15,'normal'))
    password.place(x=600,y=420)


    address = Entry(createacc_screen,textvariable = address_var,
font=('calibre',15,'normal'))
    address.place(x=600,y=520)


    phone = Entry(createacc_screen,textvariable = phone_var,
font=('calibre',15,'normal'))
    phone.place(x=600,y=620)



    but1=Button(createacc_screen,text="SUBMIT", bg='beige', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=check)
    but1.place(x=1200, y=600, anchor="center")
    butback=Button(createacc_screen,text="BACK", bg='RED', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(home,createacc_
screen.destroy))
    butback.place(x=1200, y=700, anchor="center")
```

Python script with the import psycopg2 statement, which imports the psycopg2 module, is provided.

Other modules like tkinter, PIL, tkinter.messagebox, and tkinter.ttk are also imported by the script. These modules are employed in the development of GUI applications.

Using the psycopg2.connect() method, the script creates a connection to a PostgreSQL database. It details the username, host, password, and database name for the connection.

The script defines a number of functions, such as home(), createacc(), and combine_funcs().

A new function that combines the functionality of the supplied functions is returned by the combine_funcs() function, which accepts multiple functions as parameters.

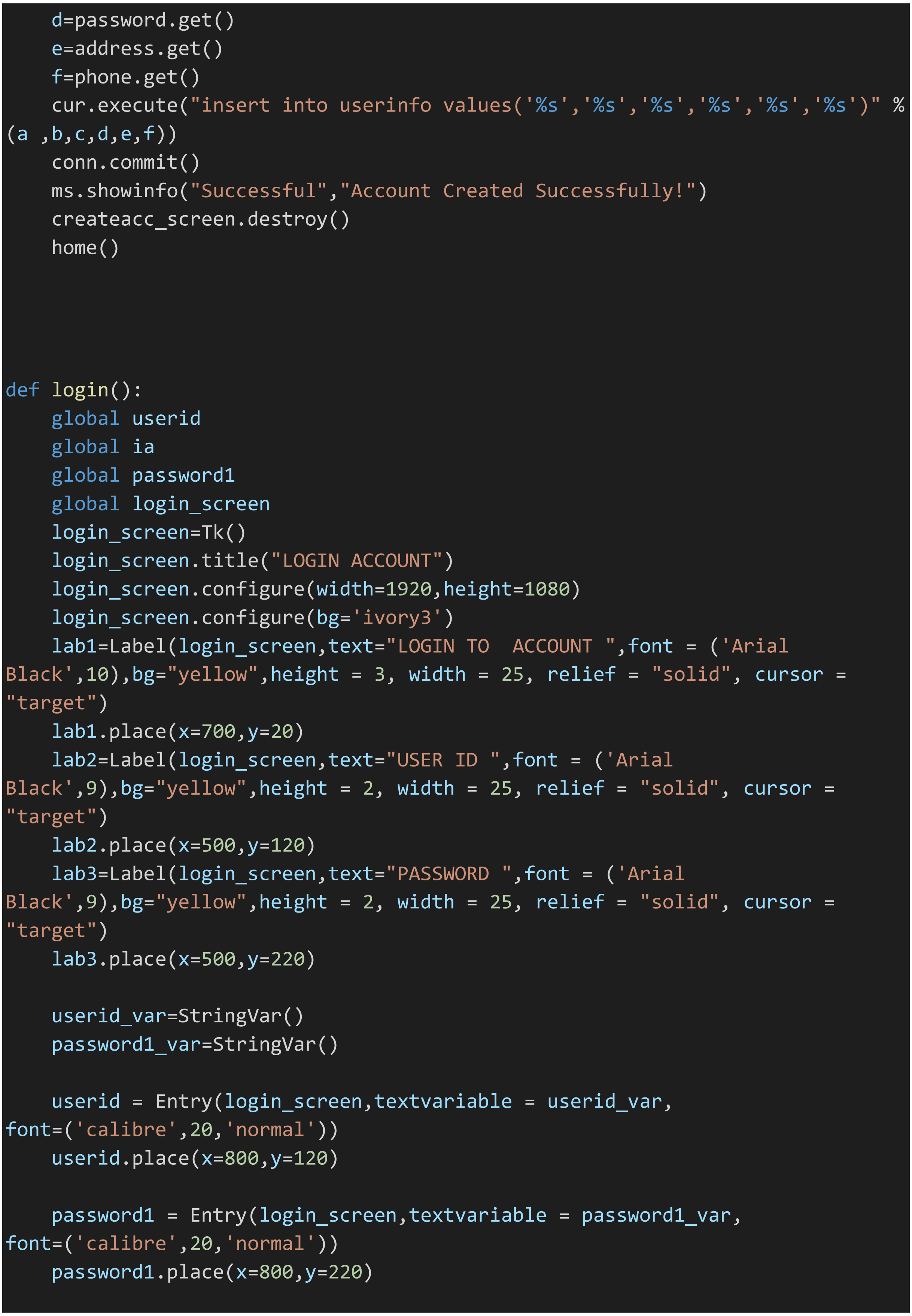
The Tk() class from the tkinter module is used to build a GUI window by the home() function. It determines the window's size, colour, and title. Additionally, it makes buttons and labels for the GUI.

Another GUI window for creating a user account is created by the createacc() function. It determines the window's size, colour, and title. For entering user data, it creates labels, entry fields, and buttons.

In order to construct the GUI elements and manage user interactions, the script additionally creates additional utility functions and makes use of numerous widgets and methods made available by the tkinter module.

It should be noted that the provided code is lacking in certain areas, including the implementation of the check() function and the needed imports.

```


def check():

    cur.execute("select uid from userinfo ")
    a1=cur.fetchall()
    for i in a1:
        (y,)=i
        if uid.get()==y:
            ms.showerror('Oops!','User ID Already Exists.')
            cur.execute("ROLLBACK")
            createacc()
            break
    a=name.get()
    b=uid.get()
    c=username.get()
```



```
    d=password.get()
    e=address.get()
    f=phone.get()
    cur.execute("insert into userinfo values('%s','%s','%s','%s','%s','%s')" %
(a ,b,c,d,e,f))
    conn.commit()
    ms.showinfo("Successful","Account Created Successfully!")
    createacc_screen.destroy()
    home()




def login():
    global userid
    global ia
    global password1
    global login_screen
    login_screen=Tk()
    login_screen.title("LOGIN ACCOUNT")
    login_screen.configure(width=1920,height=1080)
    login_screen.configure(bg='ivory3')
    lab1=Label(login_screen,text="LOGIN TO  ACCOUNT ",font = ('Arial
Black',10),bg="yellow",height = 3, width = 25, relief = "solid", cursor =
"target")
    lab1.place(x=700,y=20)
    lab2=Label(login_screen,text="USER ID ",font = ('Arial
Black',9),bg="yellow",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab2.place(x=500,y=120)
    lab3=Label(login_screen,text="PASSWORD ",font = ('Arial
Black',9),bg="yellow",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab3.place(x=500,y=220)

    userid_var=StringVar()
    password1_var=StringVar()

    userid = Entry(login_screen,textvariable = userid_var,
font=('calibre',20,'normal'))
    userid.place(x=800,y=120)

    password1 = Entry(login_screen,textvariable = password1_var,
font=('calibre',20,'normal'))
    password1.place(x=800,y=220)
    but1=Button(login_screen,text="LOGIN", bg='slategray2', font=("Arial 
Black", 10),height="2", 
width="25",borderwidth=1,relief="sunken",command=check1)
    but1.place(x=800, y=350, anchor="center")

    but1=Button(login_screen,text="LOGOUT", bg='red', font=("Arial Black", 
10),height="2", 
width="25",borderwidth=1,relief="sunken",command=combine_funcs(home,login_scre
en.destroy))

    but1.place(x=1200, y=600, anchor="center")
    ia=userid.get()

def check1():
    cur.execute("select uid,password from userinfo")
    var=cur.fetchall()
    for i in var:
        (y,z)=i
        if userid.get()==y and password1.get()==z:
            ms.showinfo("Successful LOGIN","Successful LOGIN")
            afterlogin()
            break

    if userid.get() != y :
       ms.showerror('Oops!','Username Not Found.')
       cur.execute("ROLLBACK")
       login_screen.destroy()
       login()


    if password1.get()!=z:
        ms.showerror('Oops!','Incorrect Password.')
        cur.execute("ROLLBACK")
        login_screen.destroy()
        login()
```

check(): This function is called when the user submits the form for creating a new account. It performs the following tasks:

- Executes a SQL query to retrieve all user IDs from the userinfo table.
- Fetches all the results returned by the query and iterates over them.
- Compares the entered user ID (uid) with each user ID from the database.
- If a match is found, it displays an error message using messagebox.showerror() indicating that the user ID already exists. It then rolls back the database transaction, calls the createacc() function to recreate the account creation screen, and breaks out of the loop.

- If no match is found, it retrieves the values entered for name (name), user ID (uid), username (username), password (password), address (address), and phone number (phone) from the corresponding Entry fields.
- Executes an SQL INSERT statement to insert the user information into the userinfo table.
- Commits the changes to the database using conn.commit().
- Displays a success message using messagebox.showinfo() to indicate that the account has been created successfully.
- Destroys the account creation screen (createacc_screen) and calls the home() function to go back to the home screen.

login(): This function is called when the user clicks on the "LOGIN" button. It performs the following tasks:

- Creates a new window for the login screen using Tk().
- Creates labels and entry fields for entering the user ID and password.
- Defines a function check1() which will be called when the user clicks on the "LOGIN" button.
- Creates a "LOGIN" button that calls the check1() function.
- Creates a "LOGOUT" button that calls the combine_funcs(home, login_screen.destroy) function to log out and destroy the login screen.
- Retrieves the user ID entered by the user (userid).

check1(): This function is called when the user clicks on the "LOGIN" button in the login screen. It performs the following tasks:

- Executes a SQL query to retrieve all user IDs and passwords from the userinfo table.
- Fetches all the results returned by the query and iterates over them.
- Compares the entered user ID (userid) and password (password1) with each user ID and password from the database.
- If a match is found, it displays a success message using messagebox.showinfo() to indicate that the login was successful. It then calls the afterlogin() function.
- If the user ID is not found, it displays an error message indicating that the username is not found. It rolls back the database transaction, destroys the login screen, and calls the login() function to recreate the login screen.
- If the password is incorrect, it displays an error message indicating that the password is incorrect. It rolls back the database transaction, destroys the login screen, and calls the login() function to recreate the login screen.

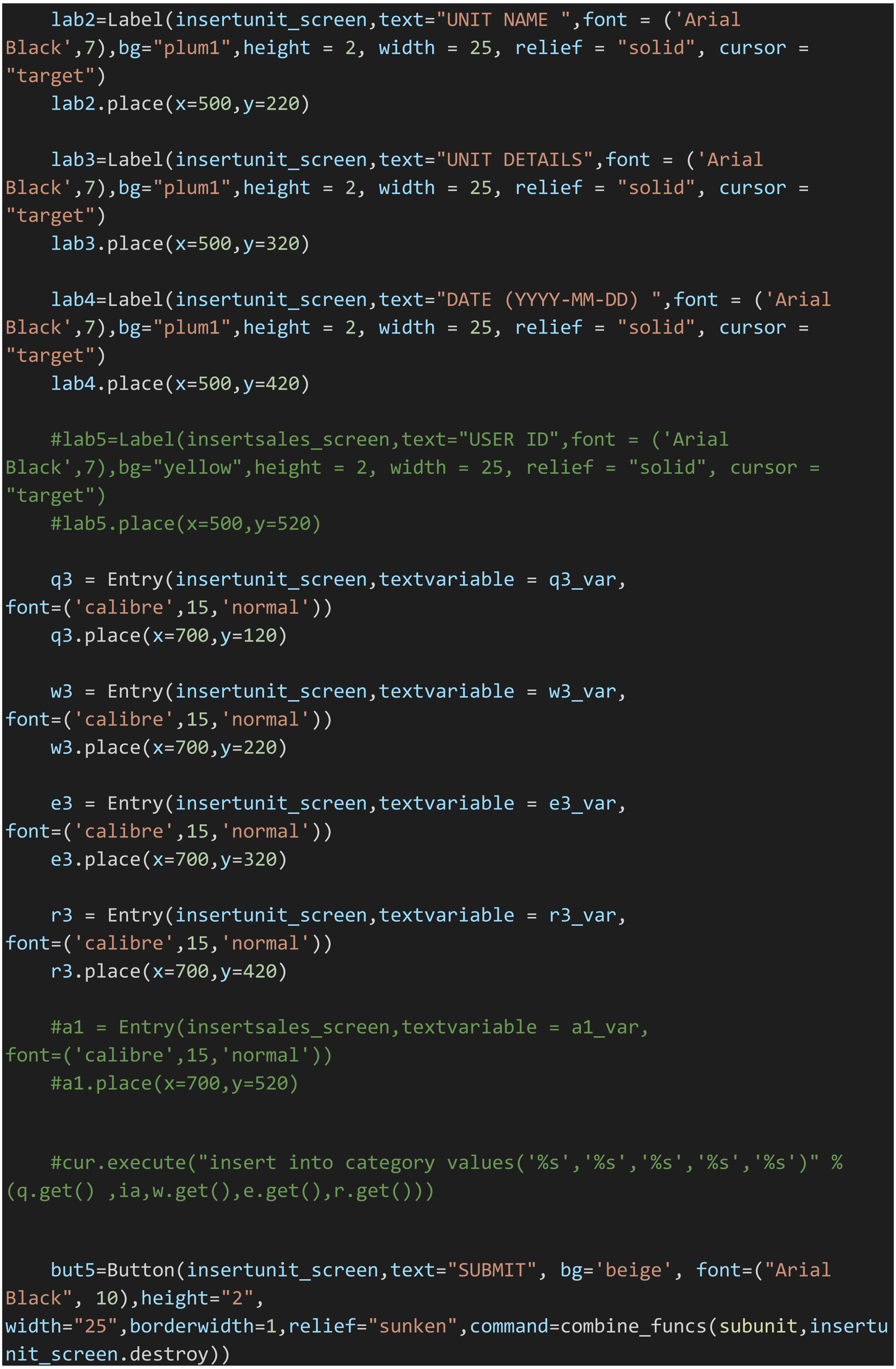
```
def afterlogin():
    global afterlogin_screen
    afterlogin_screen=Tk()
    afterlogin_screen.title("AFTER LOGIN SCREEN")
    afterlogin_screen.configure(width=1920,height=1080)
    afterlogin_screen.configure(bg='ivory3')
    lab3q=Label(afterlogin_screen,text="PLEASE SELECT ANY OF THE FOLLOWING FUNCTIONS ",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid", cursor = "target")
    lab3q.place(x=500,y=120)
    but1=Button(afterlogin_screen,text="INSERT DATA INTO VARIOUS TABLES", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insert,afterlogin_screen.destroy))
    but1.place(x=800-30, y=300, anchor="center")

    but2=Button(afterlogin_screen,text="DELETE DATA FROM PRODUCT TABLE", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(delete,afterlogin_screen.destroy))
    but2.place(x=800-30, y=400, anchor="center")
    but3=Button(afterlogin_screen,text="VIEW ALL DATA INSERTED OR UPDATED BY YOU", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(view,afterlogin_screen.destroy))
    but3.place(x=800-30, y=500, anchor="center")
    but4=Button(afterlogin_screen,text="LOGOUT", bg='RED', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(gohome,afterlogin_screen.destroy))
    but4.place(x=1200, y=700, anchor="center")

def insert():
    #afterlogin_screen.destroy()
    global insert_screen
    insert_screen=Tk()
    insert_screen.title("INSERT DATA SCREEN")
    insert_screen.configure(width=1920,height=1080)
    insert_screen.configure(bg='ivory3')

    lab3q=Label(insert_screen,text="PLEASE SELECT ANY OF THE FOLLOWING FUNCTIONS ",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid", cursor = "target")
    lab3q.place(x=500,y=50)
    but1=Button(insert_screen,text="INSERT DATA INTO CATEGORY", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insertcat,insert_screen.destroy))
    but1.place(x=770, y=200, anchor="center")
    but2=Button(insert_screen,text="INSERT DATA INTO SALES", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insertsales,insert_screen.destroy))
    but2.place(x=770, y=300, anchor="center")
    but3=Button(insert_screen,text="INSERT DATA INTO INOVICE", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insertinvoice,insert_screen.destroy))
    but3.place(x=770, y=400, anchor="center")
    but4=Button(insert_screen,text="INSERT DATA INTO UNIT", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insertunit,insert_screen.destroy))
    but4.place(x=770, y=500, anchor="center")
    but5=Button(insert_screen,text="INSERT DATA INTO PRODUCT", bg='beige', font=("Arial Black", 10),height="2", width="50",borderwidth=1,relief="sunken",command=combine_funcs(insertproduct,insert_screen.destroy))
    but5.place(x=770, y=600, anchor="center")
    but6=Button(insert_screen,text="GO BACK", bg='RED', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(afterlogin,insert_screen.destroy))
    but6.place(x=1200, y=700, anchor="center")










def insertcat():

    insertcat_screen=Tk()
    insertcat_screen.title("INSERT DATA SCREEN")
    insertcat_screen.configure(width=1920,height=1080)
    insertcat_screen.configure(bg='ivory3')
    global q,w,e,r,a

    #ia=userid.get()

    q_var=StringVar()
    w_var=StringVar()
    e_var=StringVar()
    r_var=StringVar()
    a_var=StringVar()

    lab3q=Label(insertcat_screen,text="PLEASE ENTER THE FOLLOWING DETAILS ",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid", cursor = "target")
    lab3q.place(x=500,y=30)

    lab1=Label(insertcat_screen,text="CATEGORY ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab1.place(x=500,y=120)

    lab2=Label(insertcat_screen,text="CATEGORY NAME ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab2.place(x=500,y=220)

    lab3=Label(insertcat_screen,text="CATEGORY DESCRIPTION ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab3.place(x=500,y=320)

    lab4=Label(insertcat_screen,text="DATE (YYYY-MM-DD) ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=420)

    #lab5=Label(insertcat_screen,text="USER ID",font = ('Arial Black',7),bg="yellow",height = 2, width = 25, relief = "solid", cursor = "target")
    #lab5.place(x=500,y=520)

    q = Entry(insertcat_screen,textvariable = q_var, font=('calibre',15,'normal'))
    q.place(x=700,y=120)

    w = Entry(insertcat_screen,textvariable = w_var, font=('calibre',15,'normal'))
    w.place(x=700,y=220)

    e = Entry(insertcat_screen,textvariable = e_var, font=('calibre',15,'normal'))
    e.place(x=700,y=320)

    r = Entry(insertcat_screen,textvariable = r_var, font=('calibre',15,'normal'))
    r.place(x=700,y=420)

    #a = Entry(insertcat_screen,textvariable = a_var, font=('calibre',15,'normal'))
    #a.place(x=700,y=520)

    #cur.execute("insert into category values('%s','%s','%s','%s','%s')" % (q.get(),ia,w.get(),e.get(),r.get()))

    but5=Button(insertcat_screen,text="SUBMIT", bg='lightblue', font=("Arial Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(subcat,insertcat_screen.destroy))
    but5.place(x=770, y=650, anchor="center")

    but6=Button(insertcat_screen,text="GO BACK", bg='red', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(insert,insertcat_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")


def subcat():

    cur.execute("select cid from category")
    for i in cur.fetchall():
        (y,)=i
        if q.get()==y :
            ms.showinfo("CATEGORY ID ALREADY EXISTS ","CATEGORY ID ALREADY EXISTS")
            cur.execute("ROLLBACK")
            insertcat()
            break

    cur.execute("insert into category values('%s','%s','%s','%s','%s')" % (q.get() ,userid.get(),w.get(),e.get(),r.get()))
    conn.commit()
    insert()




def insertsales():
    insertsales_screen=Tk()
    insertsales_screen.title("INSERT DATA SCREEN")
    insertsales_screen.configure(width=1920,height=1080)
    insertsales_screen.configure(bg='ivory3')
    global q1,w1,e1,r1,a1

    #ia=userid.get()

    q1_var=StringVar()
    w1_var=StringVar()
    e1_var=StringVar()
    r1_var=StringVar()
    a1_var=StringVar()
    lab3q=Label(insertsales_screen,text="PLEASE ENTER THE FOLLOWING DETAILS ",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid", cursor = "target")
    lab3q.place(x=500,y=30)

    lab1=Label(insertsales_screen,text="SALES ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab1.place(x=500,y=120)

    lab2=Label(insertsales_screen,text="QUANTITY ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab2.place(x=500,y=220)

    lab3=Label(insertsales_screen,text="TOTAL AMOUNT",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab3.place(x=500,y=320)

    lab4=Label(insertsales_screen,text="DATE (YYYY-MM-DD) ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=420)

    #lab5=Label(insertsales_screen,text="USER ID",font = ('Arial Black',7),bg="yellow",height = 2, width = 25, relief = "solid", cursor = "target")
    #lab5.place(x=500,y=520)

    q1 = Entry(insertsales_screen,textvariable = q1_var, font=('calibre',15,'normal'))
    q1.place(x=700,y=120)

    w1 = Entry(insertsales_screen,textvariable = w1_var, font=('calibre',15,'normal'))
    w1.place(x=700,y=220)

    e1 = Entry(insertsales_screen,textvariable = e1_var, font=('calibre',15,'normal'))
    e1.place(x=700,y=320)

    r1 = Entry(insertsales_screen,textvariable = r1_var, font=('calibre',15,'normal'))
    r1.place(x=700,y=420)

    #a1 = Entry(insertsales_screen,textvariable = a1_var, font=('calibre',15,'normal'))
    #a1.place(x=700,y=520)

    #cur.execute("insert into category values('%s','%s','%s','%s','%s')" % (q.get(),ia,w.get(),e.get(),r.get()))

    but5=Button(insertsales_screen,text="SUBMIT", bg='lightblue', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(subsales,insertsales_screen.destroy))
    but5.place(x=770, y=550, anchor="center")

    but6=Button(insertsales_screen,text="GO BACK", bg='red', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(insert,insertsales_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")

def subsales():

    cur.execute("select sid from sales")
    for i in cur.fetchall():
        (y,)=i
        if q1.get()==y :
            ms.showinfo("SALES ID ALREADY EXISTS ","SALES ID ALREADY EXISTS")
            cur.execute("ROLLBACK")
            conn.commit()
            insertsales()
            break

    cur.execute("insert into sales values('%s','%s','%s','%s','%s')" % (q1.get(),w1.get(),e1.get(),r1.get(),userid.get()))
    conn.commit()
    insert()

```

- The afterlogin() function creates a GUI screen after the user logs in. It creates a window using the Tkinter library and configures its properties such as title, size, and background color. It also adds a label and several buttons with different functionalities to the screen.
- The insert() function is another GUI screen that allows the user to insert data into various tables. It creates a new window and configures its properties. It adds a label and several buttons for different insertion options.
- The insertcat() function is specific to inserting data into the category table. It creates a new window and configures its properties. It adds labels and entry fields for the

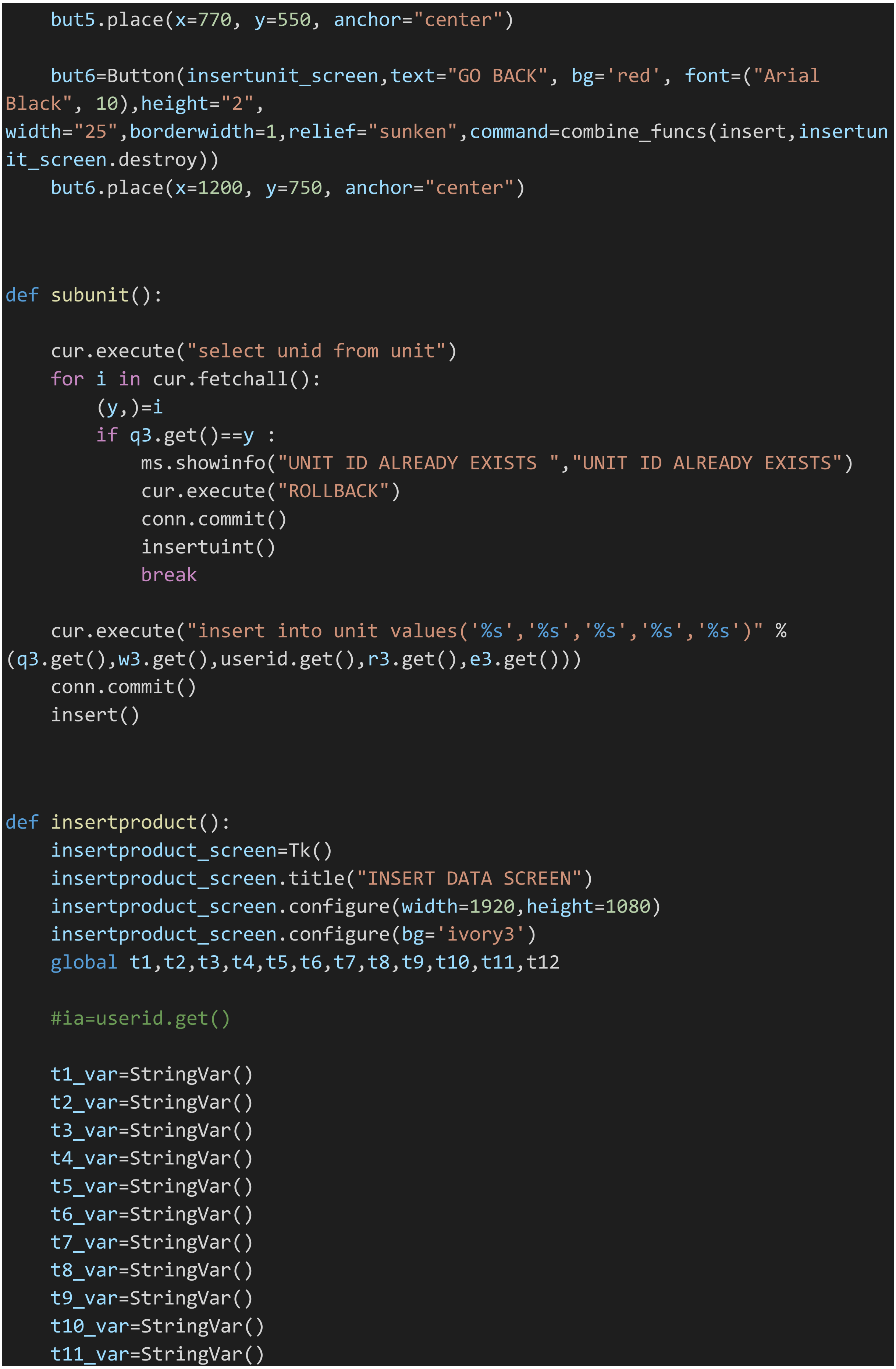
user to enter category details. It also includes buttons for submitting the data or returning to the previous screen.

- The subcat() function is called when the user submits category data. It performs some validation by checking if the category ID already exists in the category table. If it exists, it shows an info message and rolls back the database transaction. Otherwise, it inserts the data into the category table and commits the transaction.
- The insertsales() function is similar to insertcat() but specific to inserting data into the sales table. It creates a new window, configures its properties, and includes labels and entry fields for the user to enter sales details.
- The subsales() function is called when the user submits sales data. It performs a similar validation as subcat() by checking if the sales ID already exists in the sales table. If it exists, it shows an info message and rolls back the transaction. Otherwise, it inserts the data into the sales table and commits the transaction.

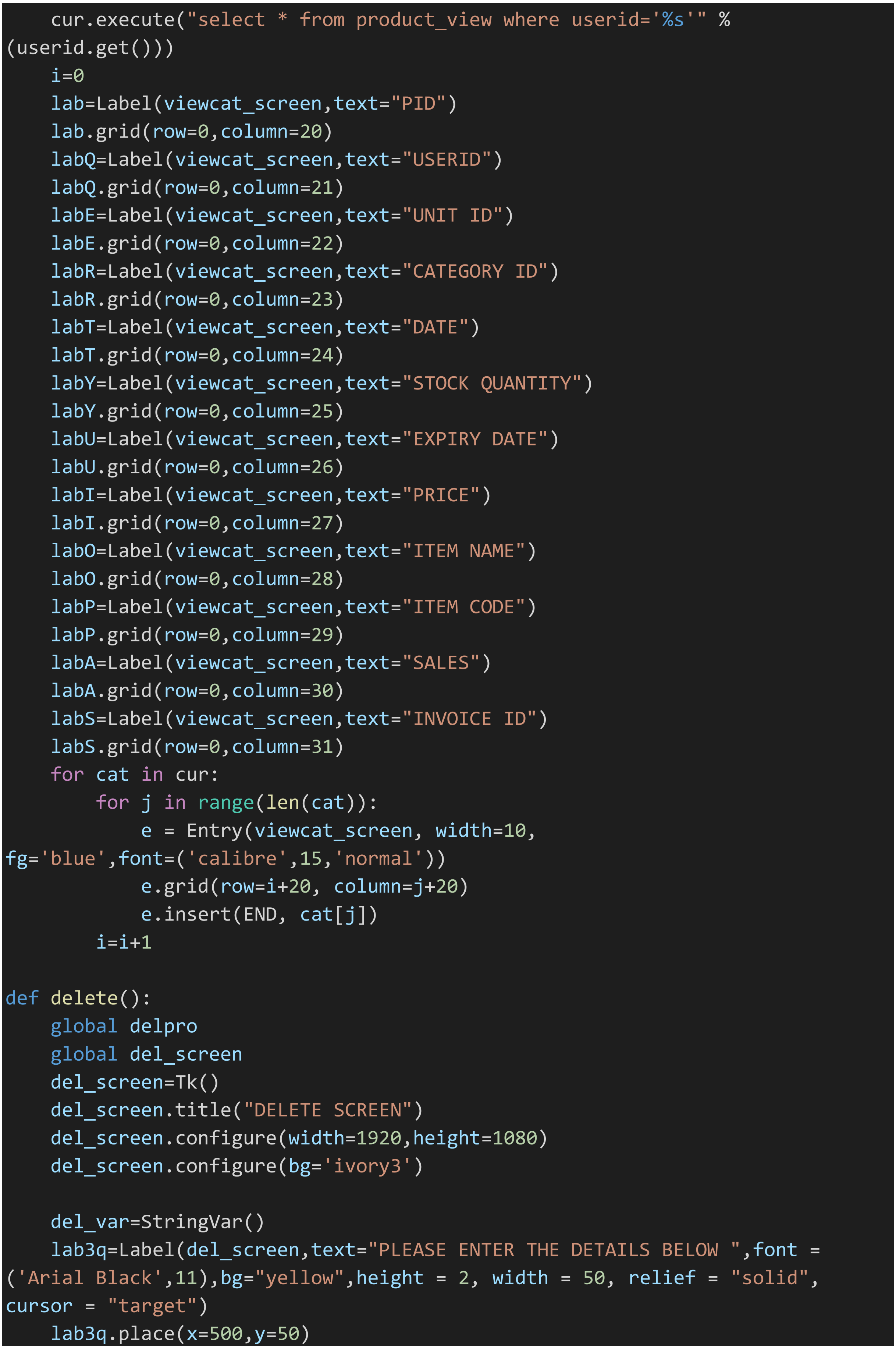
```
def insertinvoice():
    insertinvoice_screen=Tk()
    insertinvoice_screen.title("INSERT DATA SCREEN")
    insertinvoice_screen.configure(width=1920,height=1080)
    insertinvoice_screen.configure(bg='ivory3')
    global q2,w2,e2,r2,a2

    #ia=userid.get()

    q2_var=StringVar()
    w2_var=StringVar()
    e2_var=StringVar()
    r2_var=StringVar()
    a2_var=StringVar()
    lab3q=Label(insertinvoice_screen,text="PLEASE ENTER THE FOLLOWING DETAILS
",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief =
"solid", cursor = "target")
    lab3q.place(x=500,y=30)

    lab1=Label(insertinvoice_screen,text="INVOICE ID",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab1.place(x=500,y=120)

    lab2=Label(insertinvoice_screen,text="DISCOUNT PRICE ",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab2.place(x=500,y=220)
    lab3=Label(insertinvoice_screen,text="TOTAL AMOUNT",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab3.place(x=500,y=320)

    lab4=Label(insertinvoice_screen,text="DATE (YYYY-MM-DD) ",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab4.place(x=500,y=420)

    #lab5=Label(insertsales_screen,text="USER ID",font = ('Arial
Black',7),bg="yellow",height = 2, width = 25, relief = "solid", cursor =
"target")
    #lab5.place(x=500,y=520)

    q2 = Entry(insertinvoice_screen,textvariable = q2_var,
font=('calibre',15,'normal'))
    q2.place(x=700,y=120)

    w2 = Entry(insertinvoice_screen,textvariable = w2_var,
font=('calibre',15,'normal'))
    w2.place(x=700,y=220)

    e2 = Entry(insertinvoice_screen,textvariable = e2_var,
font=('calibre',15,'normal'))
    e2.place(x=700,y=320)

    r2 = Entry(insertinvoice_screen,textvariable = r2_var,
font=('calibre',15,'normal'))
    r2.place(x=700,y=420)

    #a1 = Entry(insertsales_screen,textvariable = a1_var,
font=('calibre',15,'normal'))
    #a1.place(x=700,y=520)


    #cur.execute("insert into category values('%s','%s','%s','%s','%s')" %
(q.get() ,ia,w.get(),e.get(),r.get()))


    but5=Button(insertinvoice_screen,text="SUBMIT", bg='lightblue',
font=("Arial Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(subinvoice,inse
rtinvoice_screen.destroy))
    but5.place(x=770, y=550, anchor="center")

    but6=Button(insertinvoice_screen,text="GO BACK", bg='red', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(insert,insertin
voice_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")

def subinvoice():

    cur.execute("select iid from invoice")
    for i in cur.fetchall():
        (y,)=i
        if q2.get()==y :
            ms.showinfo("INVOICE ID ALREADY EXISTS ","INVOICE ID ALREADY
EXISTS")
            cur.execute("ROLLBACK")
            conn.commit()
            insertinvoice()
            break

    cur.execute("insert into invoice values('%s','%s','%s','%s','%s')" %
(q2.get(),r2.get(),w2.get(),e2.get(),userid.get()))
    conn.commit()
    insert()

def insertunit():
    insertunit_screen=Tk()
    insertunit_screen.title("INSERT DATA SCREEN")
    insertunit_screen.configure(width=1920,height=1080)
    insertunit_screen.configure(bg='ivory3')
    global q3,w3,e3,r3,a3

    #ia=userid.get()

    q3_var=StringVar()
    w3_var=StringVar()
    e3_var=StringVar()
    r3_var=StringVar()
    a3_var=StringVar()
    lab3q=Label(insertunit_screen,text="PLEASE ENTER THE FOLLOWING DETAILS
",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief =
"solid", cursor = "target")
    lab3q.place(x=500,y=30)

    lab1=Label(insertunit_screen,text="UNIT ID",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab1.place(x=500,y=120)
```



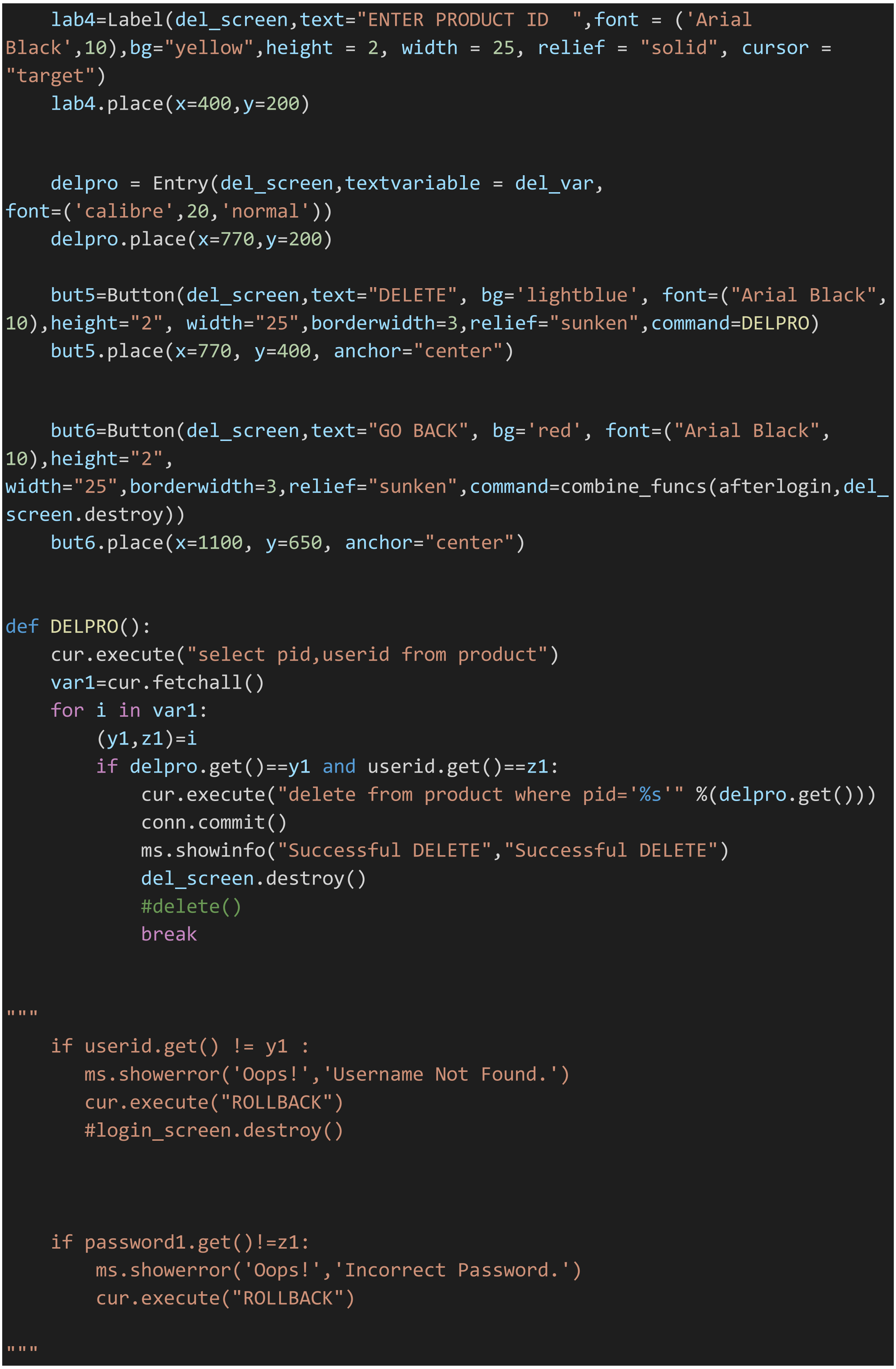
```
    lab2=Label(insertunit_screen,text="UNIT NAME ",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab2.place(x=500,y=220)

    lab3=Label(insertunit_screen,text="UNIT DETAILS",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab3.place(x=500,y=320)

    lab4=Label(insertunit_screen,text="DATE (YYYY-MM-DD) ",font = ('Arial
Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab4.place(x=500,y=420)

    #lab5=Label(insertsales_screen,text="USER ID",font = ('Arial
Black',7),bg="yellow",height = 2, width = 25, relief = "solid", cursor =
"target")
    #lab5.place(x=500,y=520)

    q3 = Entry(insertunit_screen,textvariable = q3_var,
font=('calibre',15,'normal'))
    q3.place(x=700,y=120)

    w3 = Entry(insertunit_screen,textvariable = w3_var,
font=('calibre',15,'normal'))
    w3.place(x=700,y=220)

    e3 = Entry(insertunit_screen,textvariable = e3_var,
font=('calibre',15,'normal'))
    e3.place(x=700,y=320)

    r3 = Entry(insertunit_screen,textvariable = r3_var,
font=('calibre',15,'normal'))
    r3.place(x=700,y=420)

    #a1 = Entry(insertsales_screen,textvariable = a1_var,
font=('calibre',15,'normal'))
    #a1.place(x=700,y=520)


    #cur.execute("insert into category values('%s','%s','%s','%s','%s')" %
(q.get() ,ia,w.get(),e.get(),r.get()))


    but5=Button(insertunit_screen,text="SUBMIT", bg='beige', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(subunit,insertu
nit_screen.destroy))
```



```
    but5.place(x=770, y=550, anchor="center")

    but6=Button(insertunit_screen,text="GO BACK", bg='red', font=("Arial Black", 10),height="2", width="25",borderwidth=1,relief="sunken",command=combine_funcs(insert,insertunit_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")



def subunit():

    cur.execute("select unid from unit")
    for i in cur.fetchall():
        (y,)=i
        if q3.get()==y :
            ms.showinfo("UNIT ID ALREADY EXISTS ","UNIT ID ALREADY EXISTS")
            cur.execute("ROLLBACK")
            conn.commit()
            insertuint()
            break

    cur.execute("insert into unit values('%s','%s','%s','%s','%s')" % (q3.get(),w3.get(),userid.get(),r3.get(),e3.get()))
    conn.commit()
    insert()



def insertproduct():
    insertproduct_screen=Tk()
    insertproduct_screen.title("INSERT DATA SCREEN")
    insertproduct_screen.configure(width=1920,height=1080)
    insertproduct_screen.configure(bg='ivory3')
    global t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12

    #ia=userid.get()

    t1_var=StringVar()
    t2_var=StringVar()
    t3_var=StringVar()
    t4_var=StringVar()
    t5_var=StringVar()
    t6_var=StringVar()
    t7_var=StringVar()
    t8_var=StringVar()
    t9_var=StringVar()
    t10_var=StringVar()
    t11_var=StringVar()
    t12_var=StringVar()

    lab3q=Label(insertproduct_screen,text="PLEASE ENTER THE FOLLOWING DETAILS ",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid", cursor = "target")
    lab3q.place(x=500,y=20)



    lab1=Label(insertproduct_screen,text="PRODUCT  ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab1.place(x=500,y=100)

    lab2=Label(insertproduct_screen,text="UNIT ID ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab2.place(x=500,y=150)

    lab3=Label(insertproduct_screen,text="CATEGORY ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab3.place(x=500,y=200)

    lab4=Label(insertproduct_screen,text="DATE (YYYY-MM-DD) ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=250)

    lab4=Label(insertproduct_screen,text="STOCK QUANTITY ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=300)

    lab4=Label(insertproduct_screen,text="EXPIRY DATE ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=350)

    lab4=Label(insertproduct_screen,text="PRICE ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=400)


    lab4=Label(insertproduct_screen,text="ITEM NAME ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=450)
    lab4=Label(insertproduct_screen,text="ITEM CODE ",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=500)


    lab4=Label(insertproduct_screen,text="SALES ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=550)


    lab4=Label(insertproduct_screen,text="INVOICE ID",font = ('Arial Black',7),bg="plum1",height = 2, width = 25, relief = "solid", cursor = "target")
    lab4.place(x=500,y=600)


    #lab4=Label(insertproduct_screen,text="IMAGE  ",font = ('Arial Black',7),bg="yellow",height = 2, width = 25, relief = "solid", cursor = "target")
    #lab4.place(x=500,y=600)


    t1 = Entry(insertproduct_screen,textvariable = t1_var, font=('calibre',15,'normal'))
    t1.place(x=700,y=100)

    t2 = Entry(insertproduct_screen,textvariable = t2_var, font=('calibre',15,'normal'))
    t2.place(x=700,y=150)


    t3 = Entry(insertproduct_screen,textvariable = t3_var, font=('calibre',15,'normal'))
    t3.place(x=700,y=200)

    t4 = Entry(insertproduct_screen,textvariable = t4_var, font=('calibre',15,'normal'))
    t4.place(x=700,y=250)

    t5 = Entry(insertproduct_screen,textvariable = t5_var, font=('calibre',15,'normal'))
    t5.place(x=700,y=300)


    t6 = Entry(insertproduct_screen,textvariable = t6_var, font=('calibre',15,'normal'))
    t6.place(x=700,y=350)

    t7 = Entry(insertproduct_screen,textvariable = t7_var,
font=('calibre',15,'normal'))
    t7.place(x=700,y=400)


    t8 = Entry(insertproduct_screen,textvariable = t8_var,
font=('calibre',15,'normal'))
    t8.place(x=700,y=450)

    t9 = Entry(insertproduct_screen,textvariable = t9_var,
font=('calibre',15,'normal'))
    t9.place(x=700,y=500)

    t10 = Entry(insertproduct_screen,textvariable = t10_var,
font=('calibre',15,'normal'))
    t10.place(x=700,y=550)

    t11 = Entry(insertproduct_screen,textvariable = t11_var,
font=('calibre',15,'normal'))
    t11.place(x=700,y=600)

    #t12 = Entry(insertproduct_screen,textvariable = r3_var,
font=('calibre',15,'normal'))
    #t12.place(x=700,y=600)

    but5=Button(insertproduct_screen,text="SUBMIT", bg='beige', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(subproduct,inse
rtproduct_screen.destroy))
    but5.place(x=770, y=690, anchor="center")

    but6=Button(insertproduct_screen,text="GO BACK", bg='red', font=("Arial
Black", 10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(insert,insertpr
oduct_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")


def subproduct():

    cur.execute("select pid from product")
    for i in cur.fetchall():
        (y,)=i
        if t1.get()==y :
            ms.showinfo("PRODUCT ID ALREADY EXISTS ","PRODUCT ID ALREADY
EXISTS")
            cur.execute("ROLLBACK")
            conn.commit()
            insertprodcut()
            break


    cur.execute("insert into product 
values('{0}','{1}','{2}','{3}','{4}','{5}','{6}','{7}','{8}','{9}','{10}','{11
}')"
.format(t1.get(),userid.get(),t2.get(),t3.get(),t4.get(),t5.get(),t6.get(),t7.
get(),t8.get(),t9.get(),t10.get(),t11.get()))
    conn.commit()
    insert()







def view():

    #afterlogin_screen.destroy()
    view_screen=Tk()
    view_screen.title("VIEW DATA SCREEN")
    view_screen.configure(width=1920,height=1080)
    view_screen.configure(bg='ivory3')


    lab3q=Label(view_screen,text="PLEASE SELECT FROM THE FOLLOWING OPTIONS 
",font = ('Arial Black',11),bg="yellow",height = 2, width = 50, relief = 
"solid", cursor = "target")
    lab3q.place(x=500,y=20)


    but1=Button(view_screen,text="VIEW DATA IN CATEGORY", bg='beige', 
font=("Arial Black", 10),height="2", 
width="50",borderwidth=2,relief="sunken",command=viewcat)
    but1.place(x=770, y=200, anchor="center")
    but2=Button(view_screen,text="VIEW DATA IN SALES", bg='beige', 
font=("Arial Black", 10),height="2", 
width="50",borderwidth=2,relief="sunken",command=viewsales)
    but2.place(x=770, y=300, anchor="center")
    but3=Button(view_screen,text="VIEW DATA IN INOVICE", bg='beige', 
font=("Arial Black", 10),height="2", 
width="50",borderwidth=2,relief="sunken",command=viewinvoice)
    but3.place(x=770, y=400, anchor="center")
    but4=Button(view_screen,text="VEIW DATA IN UNIT", bg='beige', font=("Arial 
Black", 10),height="2", 
width="50",borderwidth=2,relief="sunken",command=viewuint)
    but4.place(x=770, y=500, anchor="center")
    but5=Button(view_screen,text="VIEW DATA IN PRODUCT", bg='beige',
font=("Arial Black", 10),height="2",
width="50",borderwidth=2,relief="sunken",command=viewprodcut)
    but5.place(x=770, y=600, anchor="center")


    but6=Button(view_screen,text="GO BACK", bg='red', font=("Arial Black",
10),height="2",
width="25",borderwidth=1,relief="sunken",command=combine_funcs(afterlogin,view
_screen.destroy))
    but6.place(x=1200, y=750, anchor="center")





def viewcat():

    viewcat_screen=Tk()
    viewcat_screen.title("VIEW DATA SCREEN")
    viewcat_screen.configure(width=1920,height=1080)

    viewcat_screen.configure(bg='ivory3')
    cur.execute("select * from cat_view where userid='%s'" % (userid.get()))
    i=0
    lab=Label(viewcat_screen,text="CID")
    lab.grid(row=0,column=20)
    labQ=Label(viewcat_screen,text="USERID")
    labQ.grid(row=0,column=21)
    labE=Label(viewcat_screen,text="NAME")
    labE.grid(row=0,column=22)
    labR=Label(viewcat_screen,text="DESCRIPTION")
    labR.grid(row=0,column=23)
    labT=Label(viewcat_screen,text="DATE")
    labT.grid(row=0,column=24)
    for cat in cur:
        for j in range(len(cat)):
            e = Entry(viewcat_screen, width=20,
fg='blue',font=('calibre',15,'normal'))
            e.grid(row=i+20, column=j+20)
            e.insert(END, cat[j])
        i=i+1





def viewsales():

    viewcat_screen=Tk()
    viewcat_screen.title("VIEW DATA SCREEN")
    viewcat_screen.configure(width=1920,height=1080)

    viewcat_screen.configure(bg='ivory3')
    cur.execute("select * from sales_view where userid='%s'" % (userid.get()))
    i=0
    lab=Label(viewcat_screen,text="SID")
    lab.grid(row=0,column=20)
    labQ=Label(viewcat_screen,text="QUANITY")
    labQ.grid(row=0,column=21)
    labE=Label(viewcat_screen,text="TOTAL")
    labE.grid(row=0,column=22)
    labR=Label(viewcat_screen,text="DATE")
    labR.grid(row=0,column=23)
    labT=Label(viewcat_screen,text="USER ID")
    labT.grid(row=0,column=24)
    for cat in cur:
        for j in range(len(cat)):
            e = Entry(viewcat_screen, width=20,
fg='blue',font=('calibre',15,'normal'))
            e.grid(row=i+20, column=j+20)
            e.insert(END, cat[j])
        i=i+1








def viewinvoice():


    viewinv_screen=Tk()
    viewinv_screen.title("VIEW DATA SCREEN")
    viewinv_screen.configure(width=1920,height=1080)

    viewinv_screen.configure(bg='ivory3')
    cur.execute("select * from invoice_view where userid='%s'" %
(userid.get()))
    i=0
    lab=Label(viewinv_screen,text="IID")
    lab.grid(row=0,column=20)
    labQ=Label(viewinv_screen,text="DATE")
    labQ.grid(row=0,column=21)
    labE=Label(viewinv_screen,text="DISCOUNT PRICE")
    labE.grid(row=0,column=22)
    labR=Label(viewinv_screen,text="TOTAL AMOUNT")
    labR.grid(row=0,column=23)
    labT=Label(viewinv_screen,text="USER ID")
    labT.grid(row=0,column=24)
    for cat in cur:
        for j in range(len(cat)):
            e = Entry(viewinv_screen, width=20,
fg='blue',font=('calibre',15,'normal'))
            e.grid(row=i+20, column=j+20)
            e.insert(END, cat[j])
        i=i+1



def viewuint():

    viewcat_screen=Tk()
    viewcat_screen.title("VIEW DATA SCREEN")
    viewcat_screen.configure(width=1920,height=1080)

    viewcat_screen.configure(bg='ivory3')
    cur.execute("select * from unit_view where userid='%s'" % (userid.get()))
    i=0
    lab=Label(viewcat_screen,text="UNIT ID")
    lab.grid(row=0,column=20)
    labQ=Label(viewcat_screen,text="NAME")
    labQ.grid(row=0,column=21)
    labE=Label(viewcat_screen,text="USER ID")
    labE.grid(row=0,column=22)
    labR=Label(viewcat_screen,text="DATE")
    labR.grid(row=0,column=23)
    labT=Label(viewcat_screen,text="DESCRIPTION")
    labT.grid(row=0,column=24)
    for cat in cur:
        for j in range(len(cat)):
            e = Entry(viewcat_screen, width=20,
fg='blue',font=('calibre',15,'normal'))
            e.grid(row=i+20, column=j+20)
            e.insert(END, cat[j])
        i=i+1



def viewprodcut():

    viewcat_screen=Tk()
    viewcat_screen.title("VIEW DATA SCREEN")
    viewcat_screen.configure(width=1920,height=1080)

    viewcat_screen.configure(bg='ivory3')
```



```
    cur.execute("select * from product_view where userid='%s'" %
(userid.get()))
    i=0
    lab=Label(viewcat_screen,text="PID")
    lab.grid(row=0,column=20)
    labQ=Label(viewcat_screen,text="USERID")
    labQ.grid(row=0,column=21)
    labE=Label(viewcat_screen,text="UNIT ID")
    labE.grid(row=0,column=22)
    labR=Label(viewcat_screen,text="CATEGORY ID")
    labR.grid(row=0,column=23)
    labT=Label(viewcat_screen,text="DATE")
    labT.grid(row=0,column=24)
    labY=Label(viewcat_screen,text="STOCK QUANTITY")
    labY.grid(row=0,column=25)
    labU=Label(viewcat_screen,text="EXPIRY DATE")
    labU.grid(row=0,column=26)
    labI=Label(viewcat_screen,text="PRICE")
    labI.grid(row=0,column=27)
    labO=Label(viewcat_screen,text="ITEM NAME")
    labO.grid(row=0,column=28)
    labP=Label(viewcat_screen,text="ITEM CODE")
    labP.grid(row=0,column=29)
    labA=Label(viewcat_screen,text="SALES")
    labA.grid(row=0,column=30)
    labS=Label(viewcat_screen,text="INVOICE ID")
    labS.grid(row=0,column=31)
    for cat in cur:
        for j in range(len(cat)):
            e = Entry(viewcat_screen, width=10,
fg='blue',font=('calibre',15,'normal'))
            e.grid(row=i+20, column=j+20)
            e.insert(END, cat[j])
        i=i+1

def delete():
    global delpro
    global del_screen
    del_screen=Tk()
    del_screen.title("DELETE SCREEN")
    del_screen.configure(width=1920,height=1080)
    del_screen.configure(bg='ivory3')

    del_var=StringVar()
    lab3q=Label(del_screen,text="PLEASE ENTER THE DETAILS BELOW ",font =
('Arial Black',11),bg="yellow",height = 2, width = 50, relief = "solid",
cursor = "target")
    lab3q.place(x=500,y=50)
```



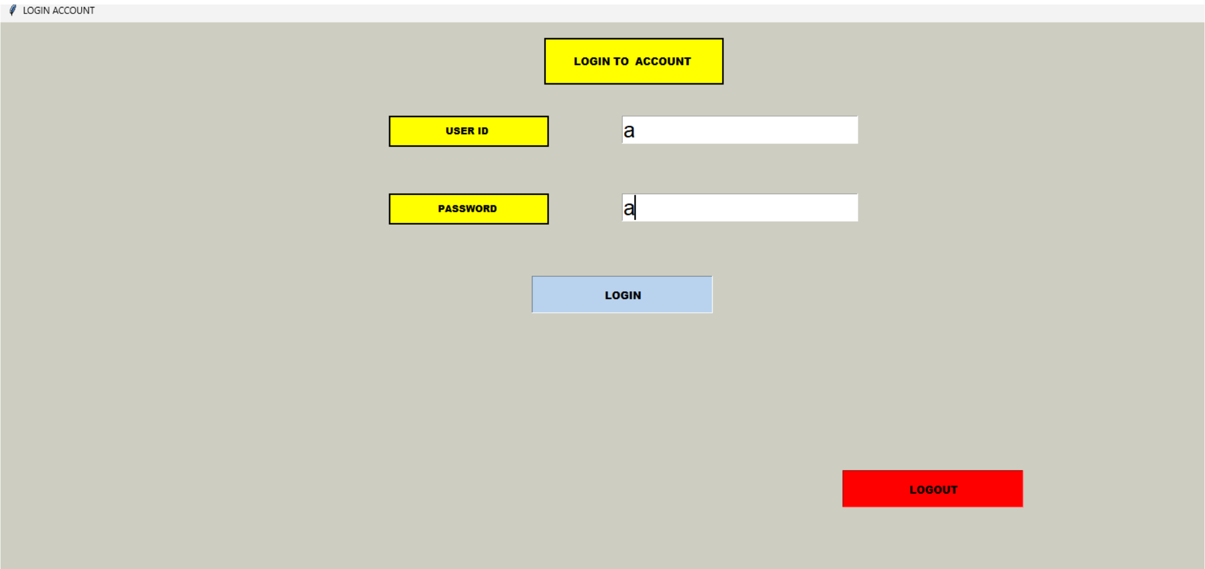
```
    lab4=Label(del_screen,text="ENTER PRODUCT ID  ",font = ('Arial
Black',10),bg="yellow",height = 2, width = 25, relief = "solid", cursor =
"target")
    lab4.place(x=400,y=200)


    delpro = Entry(del_screen,textvariable = del_var,
font=('calibre',20,'normal'))
    delpro.place(x=770,y=200)

    but5=Button(del_screen,text="DELETE", bg='lightblue', font=("Arial Black",
10),height="2", width="25",borderwidth=3,relief="sunken",command=DELPRO)
    but5.place(x=770, y=400, anchor="center")


    but6=Button(del_screen,text="GO BACK", bg='red', font=("Arial Black",
10),height="2",
width="25",borderwidth=3,relief="sunken",command=combine_funcs(afterlogin,del_
screen.destroy))
    but6.place(x=1100, y=650, anchor="center")


def DELPRO():
    cur.execute("select pid,userid from product")
    var1=cur.fetchall()
    for i in var1:
        (y1,z1)=i
        if delpro.get()==y1 and userid.get()==z1:
            cur.execute("delete from product where pid='%s'" %(delpro.get()))
            conn.commit()
            ms.showinfo("Successful DELETE","Successful DELETE")
            del_screen.destroy()
            #delete()
            break


"""
    if userid.get() != y1 :
       ms.showerror('Oops!','Username Not Found.')
       cur.execute("ROLLBACK")
       #login_screen.destroy()




    if password1.get()!=z1:
        ms.showerror('Oops!','Incorrect Password.')
        cur.execute("ROLLBACK")

"""
```

```
    #ms.showinfo("ERROR","PRODUCT NOT FOUND UNDER THE GIVEN USER ID")
    #cur.execute("ROLLBACK")
    #delete()






def gohome():
    afterlogin_screen.destroy()
    login_screen.destroy()
    home()



home()
```

- The code defines several functions: insertinvoice(), subinvoice(), insertunit(), subunit(), insertproduct(). These functions are responsible for creating different screens for inserting data and performing data insertion into corresponding database tables.
- The insertinvoice() function creates a GUI screen for inserting invoice data. It creates labels and entry fields for entering invoice details such as ID, discount price, total amount, and date. It also includes buttons for submitting the data and going back to the previous screen.
- The subinvoice() function is called when the submit button is clicked on the invoice screen. It retrieves the entered invoice ID, checks if it already exists in the database, and displays an error message if it does. Otherwise, it inserts the invoice data into the database table.
- Similarly, the insertunit() function creates a GUI screen for inserting unit data, including ID, name, details, and date. The subunit() function performs the submission and database insertion for unit data.
- The insertproduct() function creates a GUI screen for inserting product data, including various fields such as ID, unit ID, category ID, date, stock quantity, expiry date, price, item name, item code, sales ID, and invoice ID. The subproduct() function handles the submission and database insertion for product data.

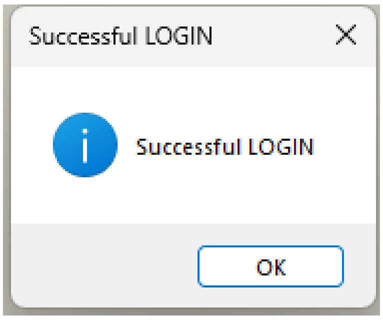
OUTPUT:

LOGIN PAGE



USER ID: a

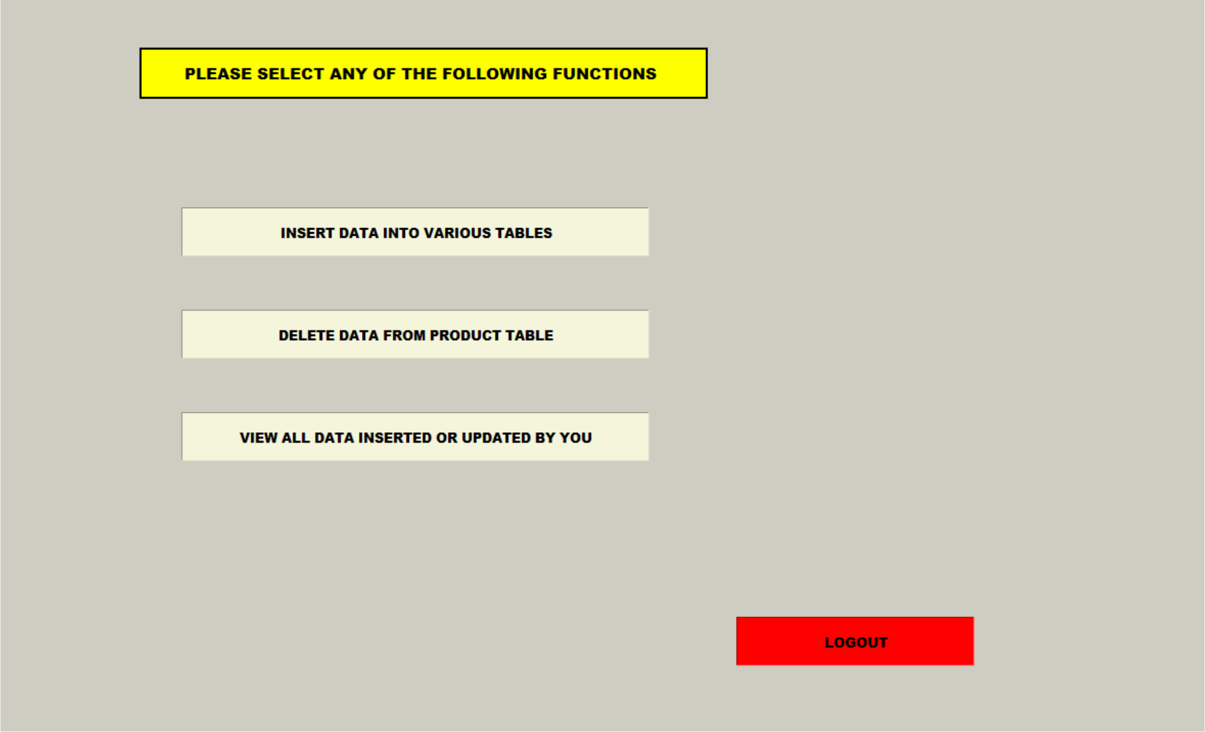
PASSWORD: a



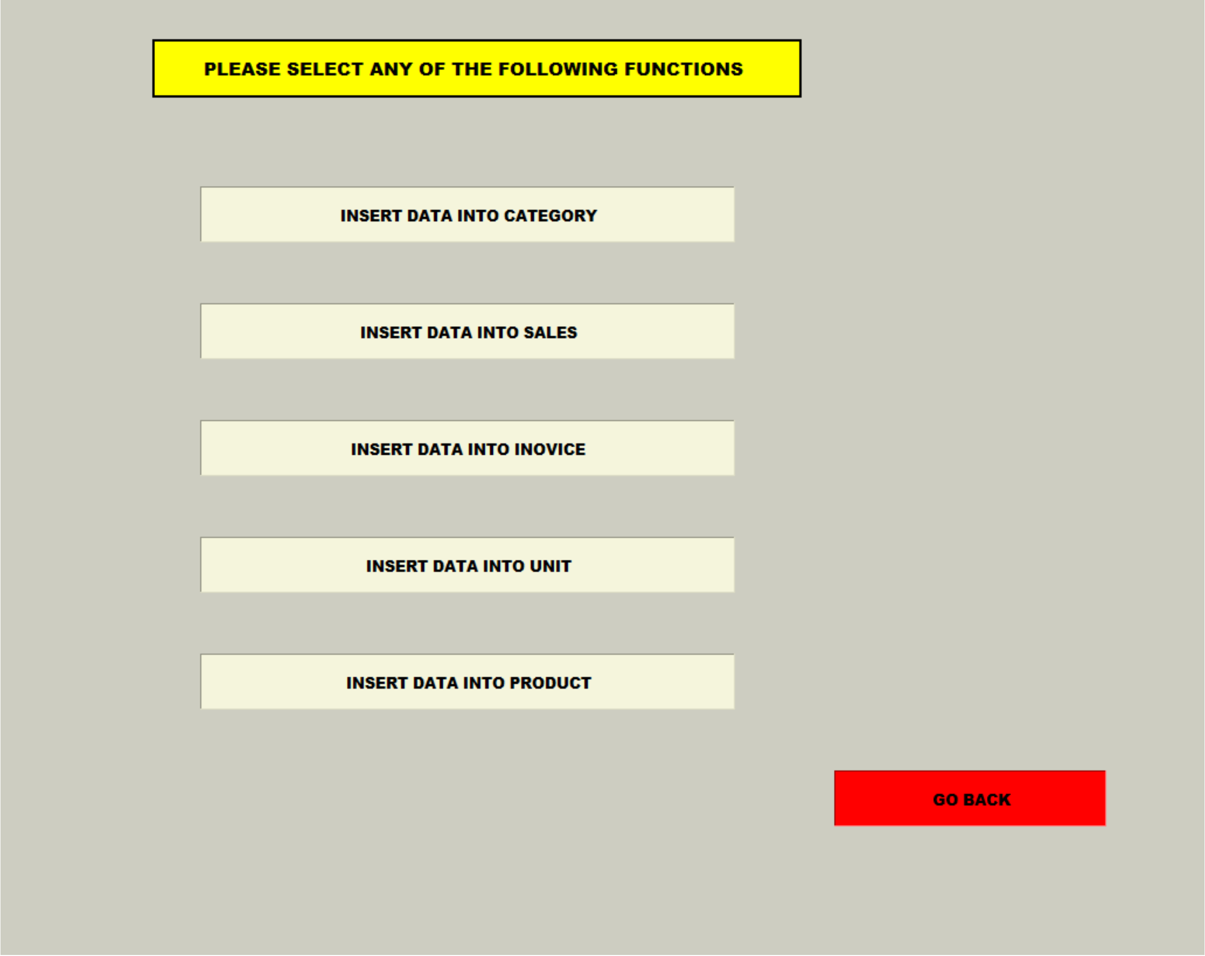
Successful LOGIN

On successful login the following prompt is presented to the user

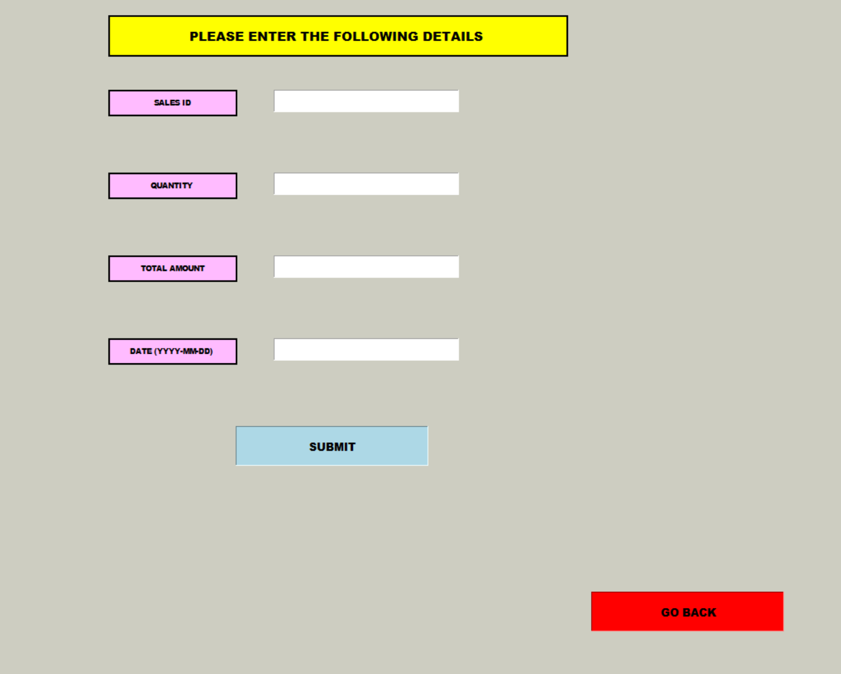
The following options for inserting, deleting and viewing data are presented to the user


PLEASE SELECT ANY OF THE FOLLOWING FUNCTIONS
INSERT DATA INTO VARIOUS TABLES
DELETE DATA FROM PRODUCT TABLE
VIEW ALL DATA INSERTED OR UPDATED BY YOU
LOGOUT

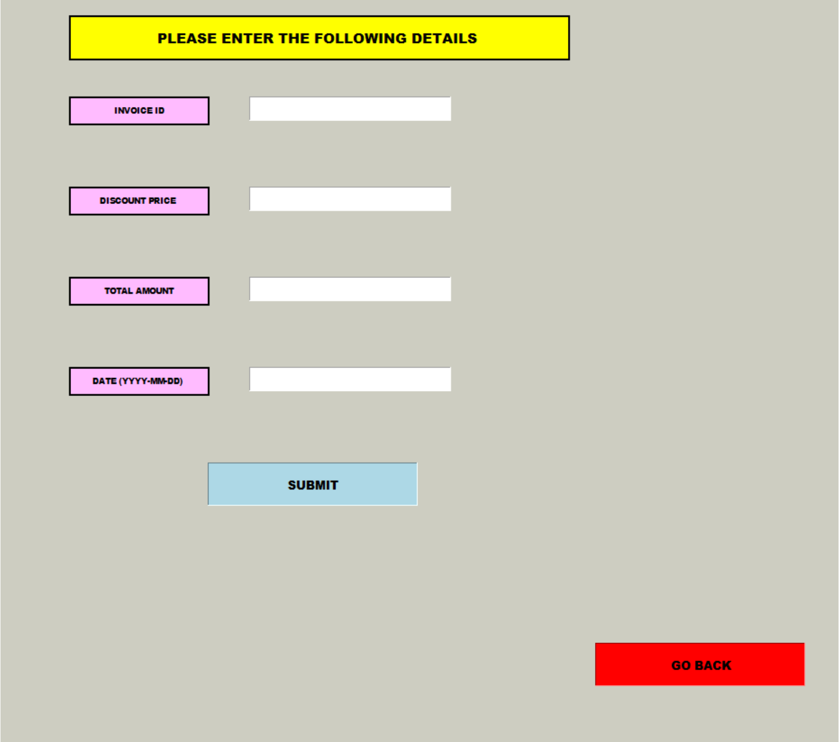
On selecting insert, the following prompt is presented



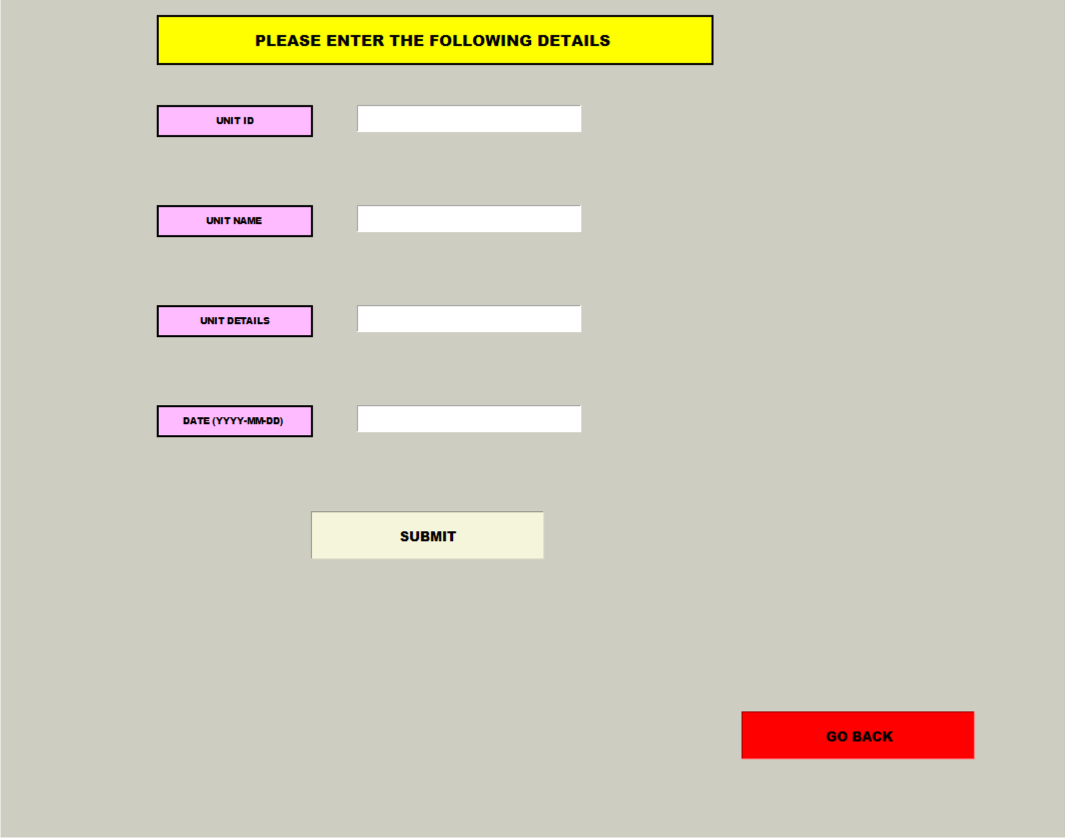
When data into sales is selected then the following options are presented



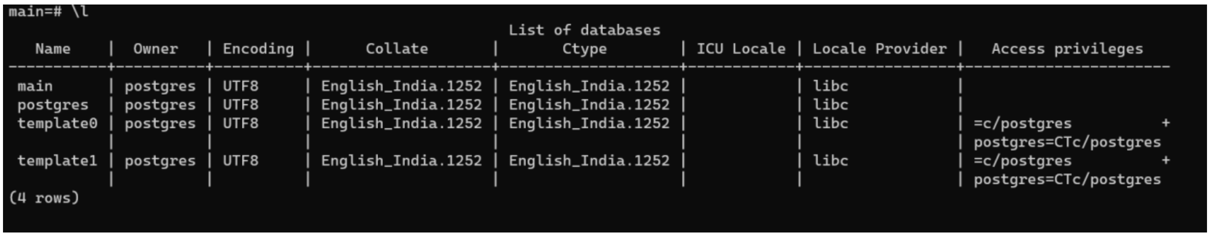
When data into invoice is selected then the following options are presented



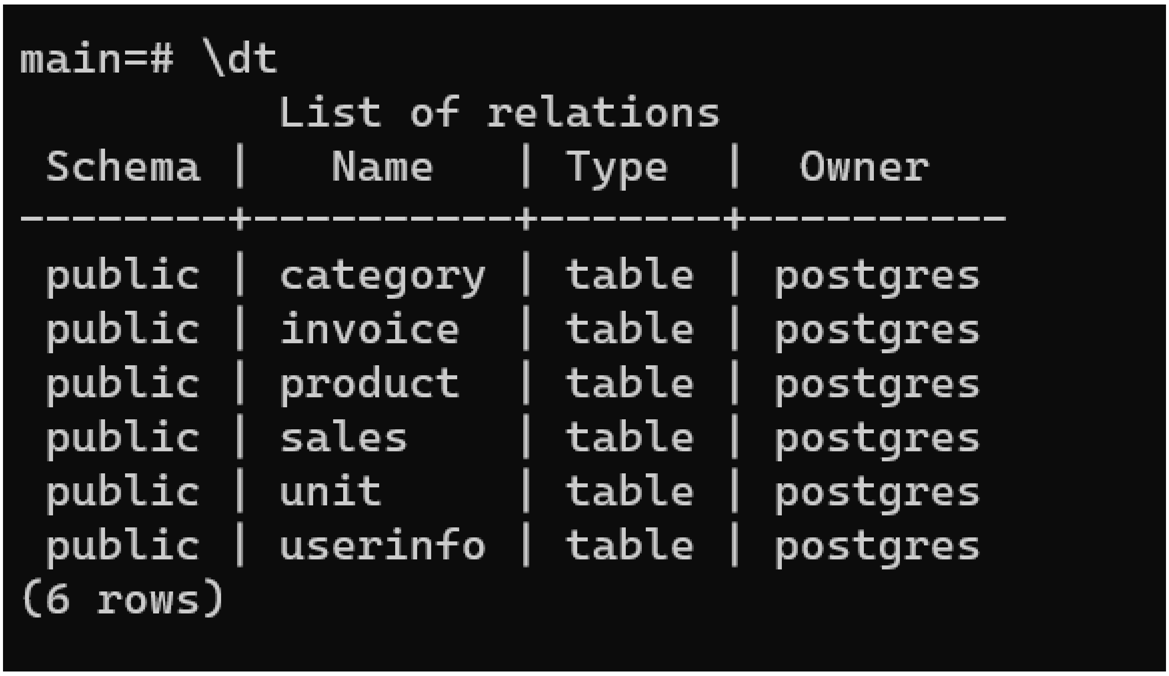
When data into unit is selected then the following options are presented



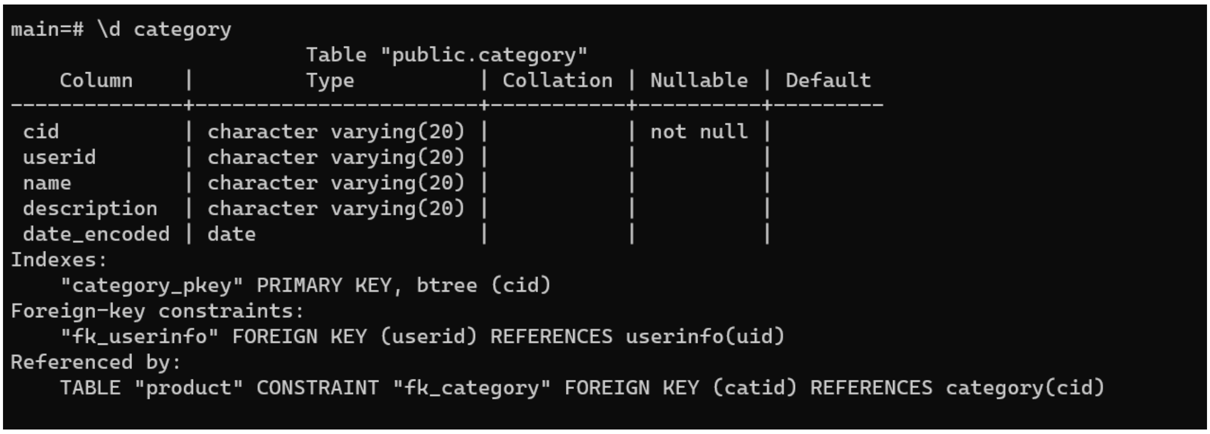
Backend data observation:



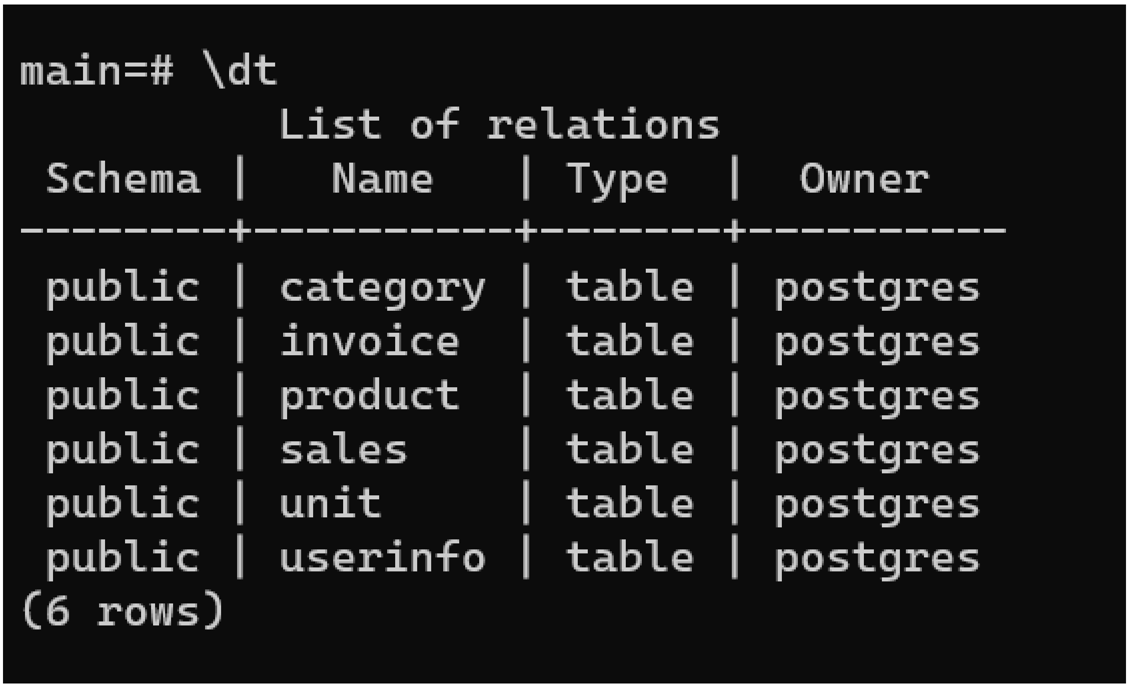
```
main=# \l
                                                     List of databases
   Name    |  Owner   | Encoding |      Collate       |       Ctype        | ICU Locale | Locale Provider |   Access privileges
-----------+----------+----------+--------------------+--------------------+------------+-----------------+-----------------------
 main      | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            |
 postgres  | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            |
 template0 | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            | =c/postgres          +
           |          |          |                    |                    |            |                 | postgres=CTc/postgres
 template1 | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            | =c/postgres          +
           |          |          |                    |                    |            |                 | postgres=CTc/postgres
(4 rows)
```



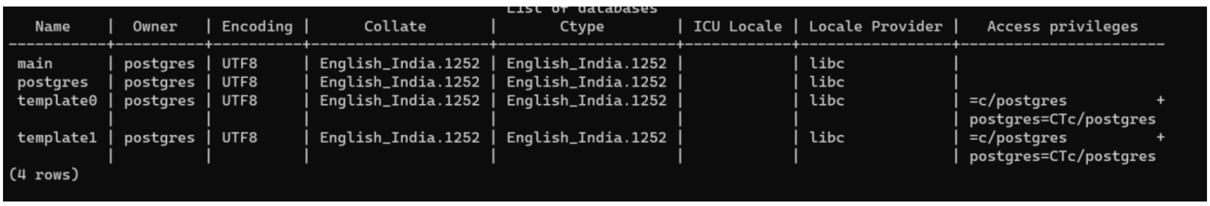
```
main=# \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | category | table | postgres
 public | invoice  | table | postgres
 public | product  | table | postgres
 public | sales    | table | postgres
 public | unit     | table | postgres
 public | userinfo | table | postgres
(6 rows)
```



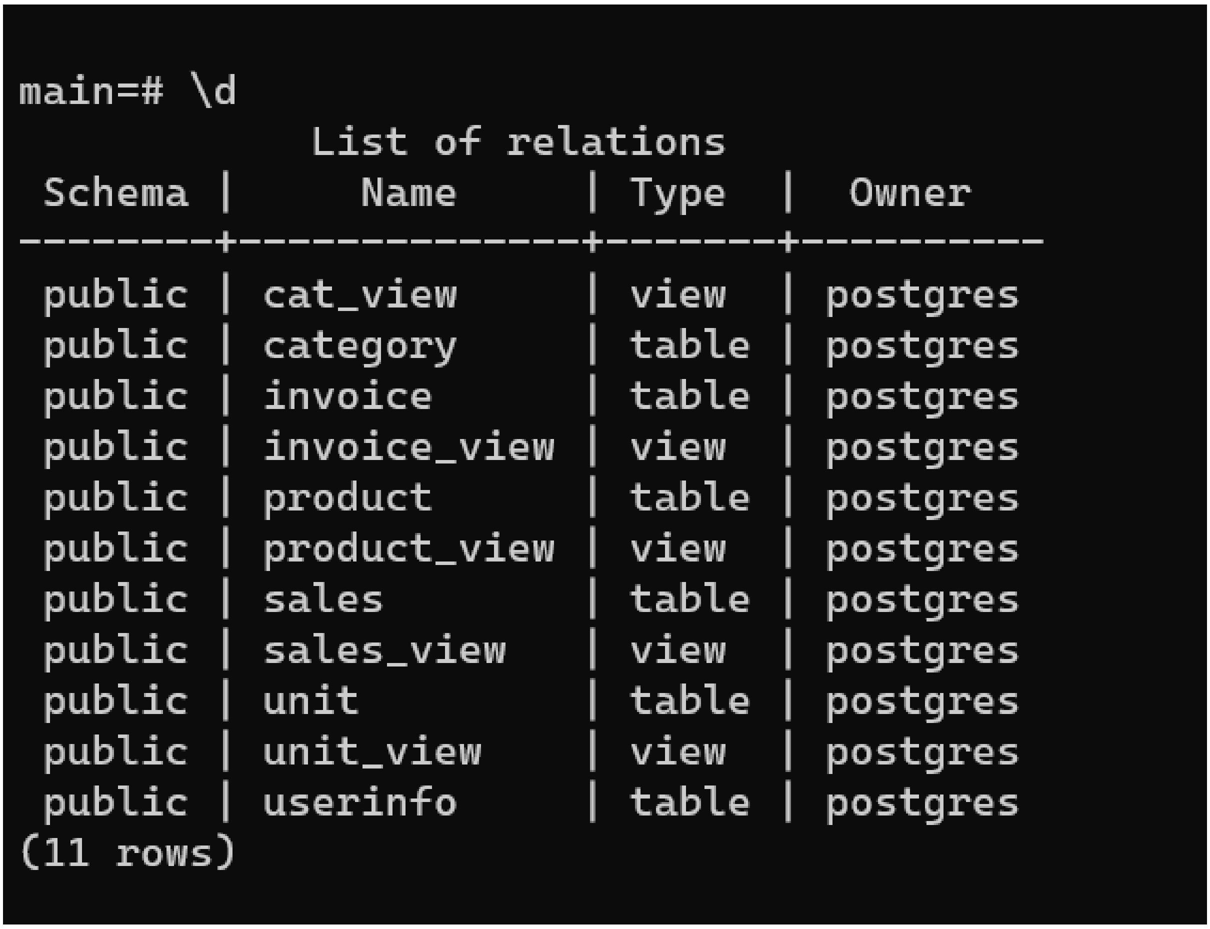
```
main=# \d category
                       Table "public.category"
    Column    |         Type          | Collation | Nullable | Default
--------------+-----------------------+-----------+----------+---------
 cid          | character varying(20) |           | not null |
 userid       | character varying(20) |           |          |
 name         | character varying(20) |           |          |
 description  | character varying(20) |           |          |
 date_encoded | date                  |           |          |
Indexes:
    "category_pkey" PRIMARY KEY, btree (cid)
Foreign-key constraints:
    "fk_userinfo" FOREIGN KEY (userid) REFERENCES userinfo(uid)
Referenced by:
    TABLE "product" CONSTRAINT "fk_category" FOREIGN KEY (catid) REFERENCES category(cid)
```



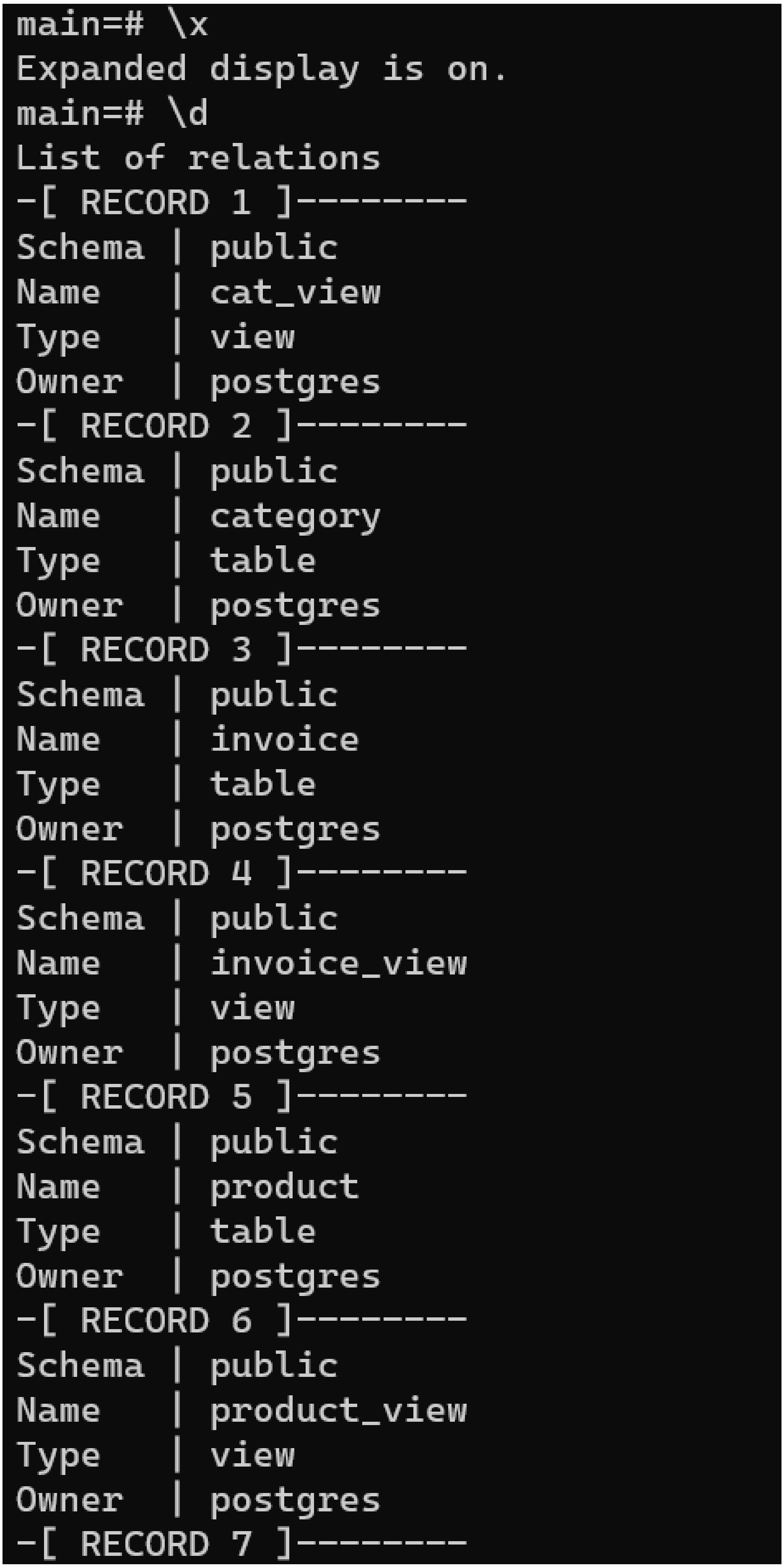
```
main=# \dt
          List of relations
 Schema |   Name   | Type  |  Owner
--------+----------+-------+----------
 public | category | table | postgres
 public | invoice  | table | postgres
 public | product  | table | postgres
 public | sales    | table | postgres
 public | unit     | table | postgres
 public | userinfo | table | postgres
(6 rows)
```



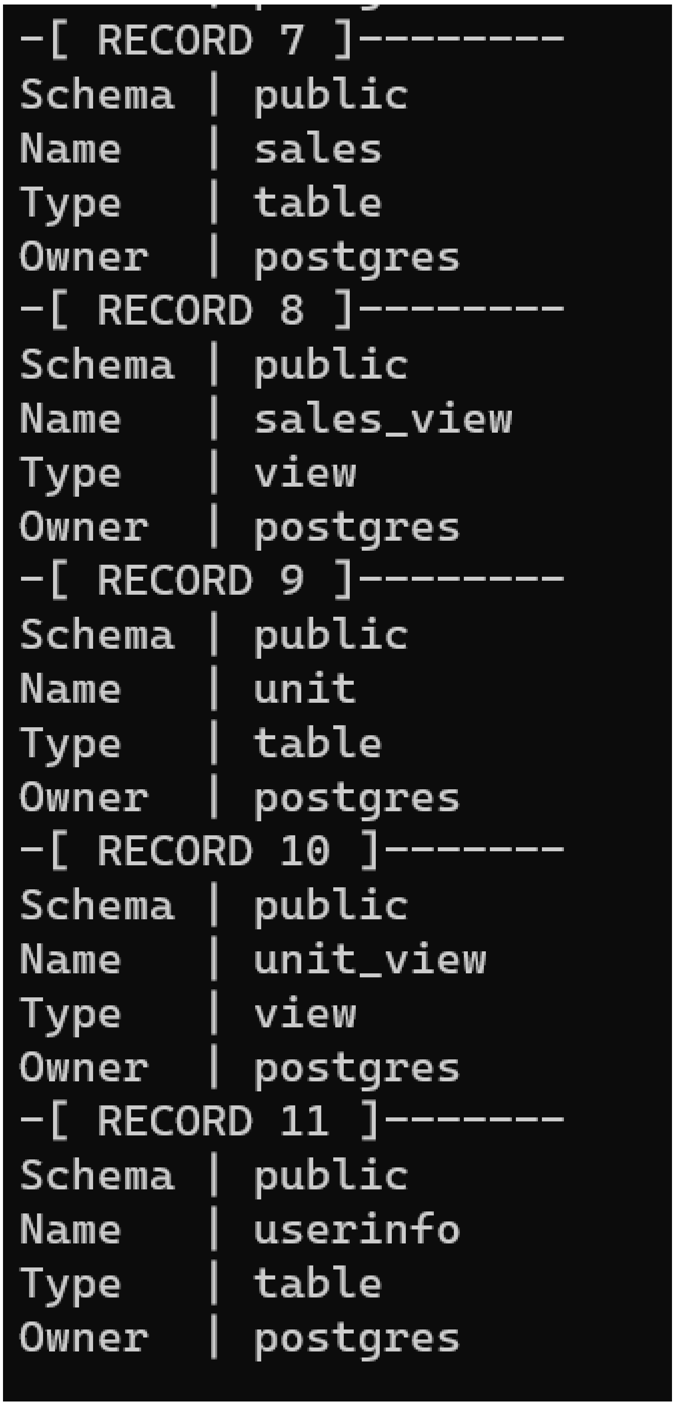
```
                                                       List of databases
   Name    |  Owner   | Encoding |      Collate       |       Ctype        | ICU Locale | Locale Provider |   Access privileges
-----------+----------+----------+--------------------+--------------------+------------+-----------------+-----------------------
 main      | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            |
 postgres  | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            |
 template0 | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            | =c/postgres          +
           |          |          |                    |                    |            |                 | postgres=CTc/postgres
 template1 | postgres | UTF8     | English_India.1252 | English_India.1252 |            | libc            | =c/postgres          +
           |          |          |                    |                    |            |                 | postgres=CTc/postgres
(4 rows)
```



```
main=# \d
            List of relations
 Schema |     Name     | Type  |  Owner
--------+--------------+-------+----------
 public | cat_view     | view  | postgres
 public | category     | table | postgres
 public | invoice      | table | postgres
 public | invoice_view | view  | postgres
 public | product      | table | postgres
 public | product_view | view  | postgres
 public | sales        | table | postgres
 public | sales_view   | view  | postgres
 public | unit         | table | postgres
 public | unit_view    | view  | postgres
 public | userinfo     | table | postgres
(11 rows)
```



```
main=# \x
Expanded display is on.
main=# \d
List of relations
-[ RECORD 1 ]--------
Schema | public
Name   | cat_view
Type   | view
Owner  | postgres
-[ RECORD 2 ]--------
Schema | public
Name   | category
Type   | table
Owner  | postgres
-[ RECORD 3 ]--------
Schema | public
Name   | invoice
Type   | table
Owner  | postgres
-[ RECORD 4 ]--------
Schema | public
Name   | invoice_view
Type   | view
Owner  | postgres
-[ RECORD 5 ]--------
Schema | public
Name   | product
Type   | table
Owner  | postgres
-[ RECORD 6 ]--------
Schema | public
Name   | product_view
Type   | view
Owner  | postgres
-[ RECORD 7 ]--------
```



```
-[ RECORD 7 ]--------
Schema | public
Name   | sales
Type   | table
Owner  | postgres
-[ RECORD 8 ]--------
Schema | public
Name   | sales_view
Type   | view
Owner  | postgres
-[ RECORD 9 ]--------
Schema | public
Name   | unit
Type   | table
Owner  | postgres
-[ RECORD 10 ]-------
Schema | public
Name   | unit_view
Type   | view
Owner  | postgres
-[ RECORD 11 ]-------
Schema | public
Name   | userinfo
Type   | table
Owner  | postgres
```