

توضیح کد:

def again1():  
 global flags  
 apt.destroy()  
 flags=''  
 again()  
again()

یک متد با نام again برنامه را آغاز میکند و اولین صفحه بالا می آید.

def again(): #for login window-----------------------------------------------------------------------------LOGIN WINDOW  
 global un, pwd, flag, root, apt  
 if flag=='apt':  
 apt.destroy()  
 root=Tk()  
 root.title('مدیریت فروشگاه')  
 Label(root,text='فروشگاه دارو').grid(row=0,column=0,columnspan=5)  
 # Label(root,text="VIT UNIVERSITY, KATPADI, VELLORE, TM").grid(row=1,column=0,columnspan=5)  
 Label(root,text='-------------------------------------------------------').grid(row=2,column=0,columnspan=5)  
 Label(root, text='نام کاربری').grid(row=3, column=0)  
 un=Entry(root,width=10)  
 un.grid(row=3, column=1)  
 Label(root, text='رمز عبور').grid(row=4, column=0)  
 pwd=Entry(root,width=10)  
 pwd.grid(row=4, column=1)  
 Button(root,width=6,text='ورود',command=check).grid(row=5, column=0)  
 Button(root,width=6,text='بستن',command=root.destroy).grid(row=5, column=1)  
 root.mainloop()

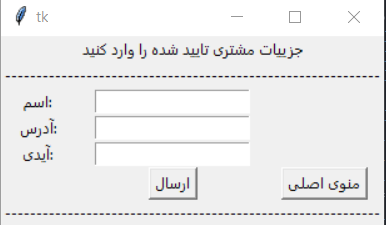
پس از شروع برنامه وارد صفحه ورود میشویم.که از کاربر نام کاربری و رمز عبور را دریافت میکنیم.

def check(): #for enter button in login window  
 global un, pwd, login, l, root  
 u=un.get()  
 p=pwd.get()  
 l.execute("select \* from log")  
 for i in l:   
 if i[0]==u and i[1]==p and u=='admin':  
 root.destroy()  
 open\_win()  
 elif i[0]==u and i[1]==p:  
 root.destroy()  
 open\_cus()  
 login.commit()

سپس به دیتابیس وصل میشویم و تمام داده های جدول log را بررسی میکنیم. اگر اطلاعات کاربر در این جدول از قبل دخیره شده بود، وارد برنامه میشویم در غیر این صورت اگر اطلاعا وارد شده اشتباه بود در همان صفحه ورود میمانیم.همچنین کاربر مینواند با فشردن دکمه بستن از برنامه خارج شود.

def open\_win(): #OPENS MAIN MENU----------------------------------------------------------------------------MAIN MENU  
 global apt, flag  
 flag='apt'  
 apt=Tk()  
 apt.title("Kosar Najafi 99221001111050")  
 Label(apt, text="فروشگاه دارو").grid(row=0,column=0,columnspan=3)  
 Label(apt, text='\*'\*80).grid(row=1,column=0,columnspan=3)  
 Label(apt, text='-'\*80).grid(row=3,column=0,columnspan=3)   
  
 Label(apt, text="ذخیره انبار").grid(row=2,column=0)  
 Button(apt,text='New V.C.', width=25, command=val\_cus).grid(row=4,column=0)  
 Button(apt,text='افزودن محصول به انبار', width=25,command=stock).grid(row=5,column=0)  
 Button(apt,text='حذف محصول از انبار', width=25,command=delete\_stock).grid(row=6,column=0)  
   
  
 Label(apt, text="دسترسی به دیتابیس").grid(row=2,column=1)  
 Button(apt,text='تغییر',width=15, command=modify).grid(row=4,column=1)  
 Button(apt,text='جستجو', width=15, command=search).grid(row=5,column=1)  
 Button(apt,text='بررسی انقضا', width=15, command=exp\_date).grid(row=6,column=1)  
  
 Label(apt, text="مدیریت جریان های مالی").grid(row=2,column=2)  
 Button(apt,text="بررسی درآمد امروز", width=20,command=show\_rev).grid(row=5,column=2)  
 Button(apt,text='صورتحساب', width=20, command=billing).grid(row=4,column=2)  
  
 Label(apt, text='-'\*80).grid(row=12,column=0,columnspan=3)   
 Button(apt,text='خروج',command=again).grid(row=13, column=2)  
 apt.mainloop()

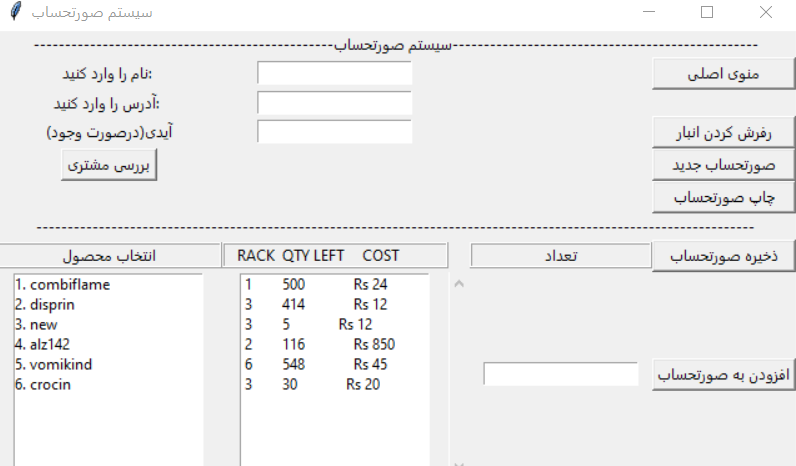
منوی اصلی نمایش داده میشود که شامل چندین دکمه مختلف است و با فشردن هر یک از آنها، وارد بخش مربوط به آن میشویم.



def val\_cus(): #to enter new valued customer-----------------------------------------------------------NEW VALUED CUSTOMER  
 global val, flag, dbt, name\_vc, add\_vc, cur, c, vc\_id  
 apt.destroy()  
 cur.execute("select \* from cus")  
 flag='val'  
 val=Tk()  
 Label(val,text="جزییات مشتری تایید شده را وارد کنید").grid(row=0,column=0,columnspan=3)  
 Label(val,text="-"\*60).grid(row=1,column=0,columnspan=3)  
 Label(val,text="اسم: ").grid(row=2,column=0)  
 name\_vc=Entry(val)  
 name\_vc.grid(row=2, column=1)  
 Label(val,text="آدرس: ").grid(row=3,column=0)  
 add\_vc=Entry(val)  
 add\_vc.grid(row=3, column=1)  
 Label(val,text="آیدی: ").grid(row=4,column=0)  
 vc\_id=Entry(val)  
 vc\_id.grid(row=4, column=1)  
 Button(val,text='ارسال',command=val\_get).grid(row=5, column=1)  
 Button(val,text='منوی اصلی',command=main\_menu).grid(row=5, column=2)  
 Label(val,text='-'\*60).grid(row=6,column=0,columnspan=3)  
 val.mainloop()

اگر دکمه مشتری جدید فشرده شود وارد یک صفحه جدید میشویم که در آن اطلاعات مشتری مثل آدرس و اسم و آیدی را وارد میکنیم تا در متغیر هایی دخیره شده و در دیتابیس ذخیره شود.

def billing(): # to create bills for customer-------------------------------------------------------------BILLING system  
 global c, cur, apt, flag, t, name, name1, add, st, names, qty, sl, qtys, vc\_id, n, namee, lb1  
 t=0  
 vc\_id=''  
 names=[]  
 qty=[]  
 sl=[]  
 n=[]  
 qtys=['']\*10  
 cur.execute("select \*from med")  
 for i in cur:  
 n.append(i[1])  
 c.commit()  
 if flag=='st':  
 st.destroy()  
 else:  
 apt.destroy()  
 flag='st'  
 st=Tk()  
 st.title('سیستم صورتحساب')  
 Label(st,text='-'\*48+'سیستم صورتحساب'+'-'\*49).grid(row=0,column=0,columnspan=7)  
 Label(st,text='نام را وارد کنید: ').grid(row=1,column=0)  
 name1=Entry(st)  
 name1.grid(row=1, column=1)  
 Label(st,text='آدرس را وارد کنید: ').grid(row=2,column=0)  
 add=Entry(st)  
 add.grid(row=2, column=1)  
 Label(st,text="آیدی(درصورت وجود)").grid(row=3, column=0)  
 vc\_id=Entry(st)  
 vc\_id.grid(row=3, column=1)  
 Button(st,text='Check V.C.', command=blue).grid(row=4, column=0)  
 Label(st,text='-'\*115).grid(row=6, column=0,columnspan=7)  
 Label(st,text='انتخاب محصول',width=25,relief='ridge').grid(row=7, column=0)  
 Label(st,text=' RACK QTY LEFT COST ',width=25,relief='ridge').grid(row=7, column=1)  
 Button(st,text='افزودن به صورتحساب',width=15,command=append2bill).grid(row=8, column=6)  
 Label(st,text='تعداد',width=20,relief='ridge').grid(row=7, column=5)  
 qtys=Entry(st)  
 qtys.grid(row=8,column=5)  
 refresh()  
 Button(st,width=15,text='منوی اصلی', command=main\_menu).grid(row=1,column=6)  
 Button(st,width=15,text='رفرش کردن انبار', command=refresh).grid(row=3,column=6)  
 Button(st,width=15,text='صورتحساب جدید', command=billing).grid(row=4,column=6)  
 Button(st,width=15,text='چاپ صورتحساب', command=print\_bill).grid(row=5,column=6)  
 Button(st,width=15,text='ذخیره صورتحساب', command=make\_bill).grid(row=7,column=6)  
   
 st.mainloop()



اگر دکمه منوی اصلی را فشار دهیم مجددا وارد صفحه منوی اصلی میشویم.با انتخاب دکمه صورتحساب وارد بخش مربوط به ایجاد صورتحساب میشویم که درآن باید ابتدا نام و آدرس و آیدی مشتری را وارد کنیم تا مشتری تایید شود که در دیتابیس وجود دارد، سپس مشتری میتواند محصولاتی را انتخاب کند و هزینه آنها در صورتحساب ثبت شود.در نهایت میتوانیم صورت حساب ایجاد شده را چاپ کنیم و انبار را رفرش کنیم.اطلاعات اضافع شده در این مرحله در دیتابیس ذخیره میشود.



def stock(): #add to stock window------------------------------------------------------------------------ADD TO STOCK  
 global cur, c, columns, accept, flag, sto, apt  
 apt.destroy()  
 flag='sto'  
 accept=['']\*10  
 sto=Tk()  
 sto.title('ورودی انبار')  
 Label(sto,text='داده محصول جدید انبار را وارد کنید').grid(row=0,column=0,columnspan=2)  
 Label(sto,text='-'\*50).grid(row=1,column=0,columnspan=2)  
 for i in range(1,len(columns)):  
 Label(sto,width=15,text=' '\*(14-len(str(columns[i])))+str(columns[i])+':').grid(row=i+2,column=0)  
 accept[i]=Entry(sto)  
 accept[i].grid(row=i+2, column=1)  
 Button(sto,width=15,text='ثبت',command=submit).grid(row=12,column=1)  
 Label(sto,text='-'\*165).grid(row=13,column=0,columnspan=7)  
 Button(sto,width=15,text='را اندازی مجدد',command=reset).grid(row=12,column=0)  
 Button(sto,width=15,text='رفرش کردن انبار',command=ref).grid(row=12,column=4)  
 for i in range(1,6):  
 Label(sto,text=columns[i]).grid(row=14,column=i-1)  
 Label(sto,text='exp rack manufacture ').grid(row=14,column=5)  
 Button(sto,width=10,text='منوی اصلی',command=main\_menu).grid(row=12,column=5)  
 ref()  
 sto.mainloop()

اگر دکمه افزودن محصول جدید به انبار را انتخاب کنیم وارد صفحه بالا میشویم و میتوانیم یک محصول جدید را با اسم و نوع و مقدار باقیمانده و هزینه و کاربرد و تاریخ انقضا و مکان قفسه و سازنده تعریف کنیم و به محصولات موجود در انبار بیفزاییم.این داده همچنین در دیتابیس نیز ذخیره میشود.

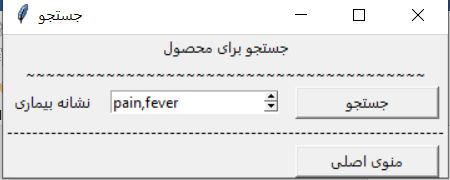


def delete\_stock(): #OPENS DELETE WINDOW----------------------------------------------------DELETES A PARTICULAR STOCK ITEM  
 global cur, c, flag, lb1, d  
 apt.destroy()  
 flag='d'  
 d=Tk()  
 d.title("حذف یک محصول از انبار")  
 Label(d,text='محصول را برای حذف شدن وارد کنید:').grid(row=0,column=0)  
 Label(d,text='',width=30,bg='white').grid(row=0,column=1)  
 Label(d,text='محصول').grid(row=2,column=0)  
 Label(d,text='تعداد. تاریخ انقضا. هزینه ').grid(row=2,column=1)  
 ren()  
 b=Button(d,width=20,text='حذف',command=delt).grid(row=0,column=3)  
 b=Button(d,width=20,text='منوی اصلی',command=main\_menu).grid(row=5,column=3)  
 d.mainloop()

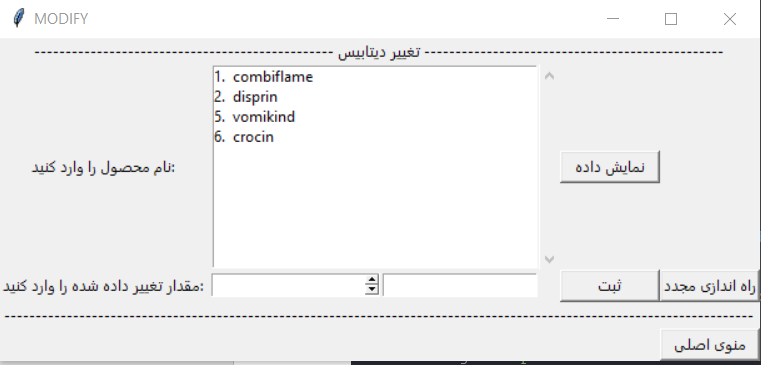


همچنین اگر روی دکمه حذف محصول از انبار در منوی اصلی کلیک کنیم، وارد صفحه بالا میشویم و میتوانیم یکی از محصولاتی که قبلا در دیتابیس ذخیره شده و اکنون در یک متغیر ذخیره شده اند را انتخاب و آن را حذف کنیم تا از انبار حذف شود بلافاصله بعد از حذف کردن از لیست نیز حذف خواهد شد.

def search(): #search window medicine and symptom details---------------------------------SEARCH MEDICINE RACK & SYMPTOMS  
 global c, cur, flag, st, mn, sym, flags  
 flag='st'  
 apt.destroy()  
 cur.execute("Select \* from med")  
 symp=['nil']  
 med\_name=['nil']  
 for i in cur:  
 symp.append(i[5])  
 med\_name.append(i[1])  
 st=Tk()  
 st.title('جستجو')  
 Label(st, text=' جستجو برای محصول ').grid(row=0, column=0,columnspan=3)  
 Label(st, text='~'\*40).grid(row=1, column=0,columnspan=3)  
 Label(st, text='نشانه بیماری').grid(row=3, column=0)  
 sym=Spinbox(st,values=symp)  
 sym.grid(row=3, column=1)  
 Button(st,width=15, text='جستجو', command=search\_med).grid(row=3, column=2)  
 Label(st, text='-'\*70).grid(row=4, column=0,columnspan=3)  
 if flags=='apt1':  
 Button(st,width=15, text='منوی اصلی', command=main\_cus).grid(row=6, column=2)  
 else:  
 Button(st,width=15, text='منوی اصلی', command=main\_menu).grid(row=6, column=2)  
 st.mainloop()

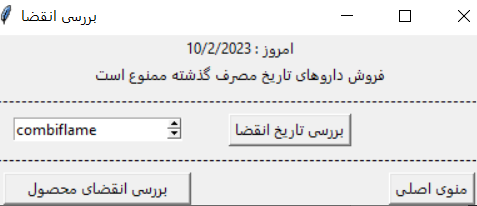


اگر روی دکمه جستجو در منوی اصلی کلیک کنیم وارد صفحه بالا میشویم و میتوانیم یک بیماری را از لیست بیماری های زخیره شده در دیتابیس انتخاب کنیم تا داروهای مربوط به آن نمایش داده شود.



def modify(): # window for modification-----------------------------------------------------------------------MODIFY  
 global cur, c, accept, flag, att, up, n, name\_, apt, st, col,col\_n  
 col=('', '', 'type', 'qty\_left', 'cost', 'purpose', 'expdt', 'loc', 'mfg')  
 col\_n=('', '', 'Type', 'Quantity Left', 'Cost', 'Purpose', 'Expiry Date', 'Rack location', 'Manufacture')  
 flag='st'  
 name\_=''  
 apt.destroy()  
 n=[]  
 cur.execute("select \* from med")  
 for i in cur:  
 n.append(i[1])  
 c.commit()  
 st=Tk()  
 st.title('MODIFY')  
 Label(st, text='-'\*48+' تغییر دیتابیس '+'-'\*48).grid(row=0, column=0,columnspan=6)  
 def onvsb(\*args):  
 name\_.yview(\*args)  
 def onmousewheel():  
 name\_.ywiew=('scroll',event.delta,'units')  
 return 'break'  
 cx=0  
 vsb=Scrollbar(orient='vertical',command=onvsb)  
 vsb.grid(row=1,column=3,sticky=N+S)  
 name\_=Listbox(st,width=43,yscrollcommand=vsb.set)  
 cur.execute("select \*from med")  
 for i in cur:  
 cx+=1  
 name\_.insert(cx,(str(i[0])+'. '+str(i[1])))  
 name\_.grid(row=1,column=1,columnspan=2)  
 c.commit()  
 name\_.bind('<MouseWheel>',onmousewheel)  
 name\_.bind('<<ListboxSelect>>', sel\_mn)  
  
 Label(st, text='نام محصول را وارد کنید: ').grid(row=1, column=0)  
 Label(st, text='مقدار تغییر داده شده را وارد کنید: ').grid(row=2, column=0)  
 att=Spinbox(st, values=col\_n)  
 att.grid(row=2, column=1)  
 up=Entry(st)  
 up.grid(row=2, column=2)  
 Button(st,width=10,text='ثبت', command=save\_mod).grid(row=2, column=4)  
 Button(st,width=10,text='راه اندازی مجدد', command=res).grid(row=2, column=5)  
 Button(st,width=10,text='نمایش داده', command=show\_val).grid(row=1, column=4)  
 Label(st, text='-'\*120).grid(row=3,column=0,columnspan=6)  
 Button(st,width=10,text='منوی اصلی',command=main\_menu).grid(row=5,column=5)  
 st.mainloop()

اگر روی دکمه تغییر در منوی اصلی کلیک کنیم وارد صفحه بالا میشویم و میتوانیم مقادیر محصولاتی که در انبار و دیتابیس ذخیره شده اند را تغییر دهیم.



def exp\_date(): # expiry window open-----------------------------------------------------------------------------EXPIRY  
 global exp, s,c, cur, flag, apt, flags  
 apt.destroy()  
 flag='exp'  
 from datetime import date  
 now=time.localtime()  
 n=[]  
 cur.execute("select \*from med")  
 for i in cur:  
 n.append(i[1])  
 c.commit()  
 exp=Tk()  
 exp.title('بررسی انقضا')  
 Label(exp,text='امروز : '+str(now[2])+'/'+str(now[1])+'/'+str(now[0])).grid(row=0, column=0, columnspan=3)  
 Label(exp,text='فروش داروهای تاریخ مصرف گذشته ممنوع است').grid(row=1, column=0,columnspan=3)  
 Label(exp,text='-'\*80).grid(row=2, column=0,columnspan=3)  
 s=Spinbox(exp,values=n)  
 s.grid(row=3, column=0)  
 Button(exp,text='بررسی تاریخ انقضا', command=s\_exp).grid(row=3, column=1)  
 Label(exp,text='-'\*80).grid(row=4, column=0,columnspan=3)  
 if flags=='apt1':  
 Button(exp,text='منوی اصلی', command=main\_cus).grid(row=5, column=2)  
 else:  
 Button(exp,width=20,text='بررسی انقضای محصول', command=exp\_dt).grid(row=5, column=0)  
 Button(exp,text='منوی اصلی', command=main\_menu).grid(row=5, column=2)  
 exp.mainloop()

اگر گزینه بررسی تاریخ انقضا در منوی اصلی را انتخاب کنیم وارد صفحه بالا میشویم و میتوانیم یک محصول را از دیتابیس انتخاب کرده و تاریخ انقضای آن را که در دیتابیس ذخیره شده است نگاه کنیم.



اگر روی دکمه بررسی درامد امروز کلیک کنیم وارد صفحه بالا میشویم و میتوانیم مقدار پولی را که امروز از فروش محصولات بدست آورده ایم مشاهده کنیم که در یک لیست و دریک جدول از دیتابیس ذخیره شده است.