

مدیریت فروشگاه

فروشگاه دارو

نام کاربری: admin

رمز عبور: admin

ورود بستن

Kosar Najafi 99221001111050

فروشگاه دارو

مدیریت جریان های مالی    دسترسی به دیتابیس    ذخیره انبار

New V.C.	تغییر	صورتحساب
افزودن محصول به انبار	جستجو	بررسی درآمد امروز
حذف محصول از انبار	بررسی انقضا	

خروج

سیستم صورتحساب

سیستم صورتحساب

نام را وارد کنید:

آدرس را وارد کنید:

Value Id (if available)

Check V.C.

منوی اصلی

رفرش کردن انبار

صورتحساب جدید

چاپ صورتحساب

انتخاب محصول	RACK	QTY	LEFT	COST	تعداد	ذخیره صورتحساب
1. combiflame	1	500		Rs 24		
2. disprin	3	414		Rs 12		
3. new	3	5		Rs 12		
4. alz142	2	116		Rs 850		
5. vomikind	6	548		Rs 45		
6. crocin	3	30		Rs 20		

افزودن به صورتحساب

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('فروشگاه مدیریت')
    Label(root,text='دارو فروشگاه').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='کاربری نام').grid(row=3, column=0)
    un=Entry(root,width=10)
    un.grid(row=3, column=1)
    Label(root, text='عبور رمز').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 ذخیره شده بود، وارد برنامه میشویم در غیر این صورت اگر اطلاعا وارد شده اشتباه بود در همان صفحه ورود میمانیم. همچنین کاربر میتواند 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="دارو فروشگاه").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)

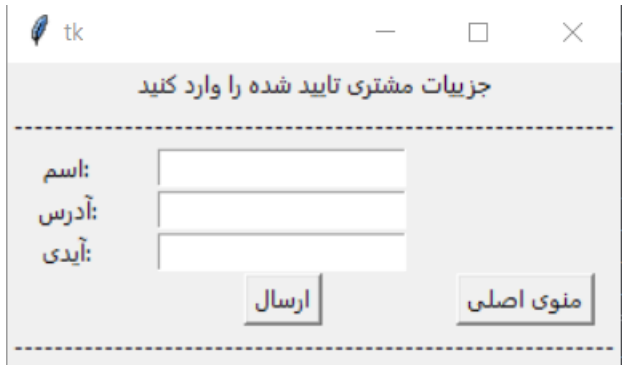
    Label(apt, text="انبار ذخیره").grid(row=2,column=0)
    Button(apt,text='New V.C.', width=25,
command=val_cus).grid(row=4,column=0)
    Button(apt,text='انبار به محصول افزودن',
width=25,command=stock).grid(row=5,column=0)
    Button(apt,text='انبار از محصول حذف',
width=25,command=delete_stock).grid(row=6,column=0)

    Label(apt, text="دیتابیس به دسترسی").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='انقضا بررسی', width=15,
command=exp_date).grid(row=6,column=1)

    Label(apt, text="مالی های جریان مدیریت").grid(row=2,column=2)
    Button(apt,text="امروز درآمد بررسی",
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="وارد شده تایید مشتری جزییات
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 ذخیره 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, lbl
    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('صورتحساب سیستم')
Label(st,text='- '*48+'صورتحساب سیستم'+'- '*49).grid(row=0,column=0,columnspan=7)
Label(st,text='کنید وارد را نام: ').grid(row=1,column=0)
name1=Entry(st)
name1.grid(row=1, column=1)
Label(st,text='کنید وارد را آدرس: ').grid(row=2,column=0)
add=Entry(st)
add.grid(row=2, column=1)
Label(st,text="(وجود درصورت) آیدی").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='محمول انتخاب',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='به افزودن',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='اصلی منوی',command=main_menu).grid(row=1,column=6)
Button(st,width=15,text='انبار کردن رفرش',command=refresh).grid(row=3,column=6)
Button(st,width=15,text='جدید صورتحساب',command=billing).grid(row=4,column=6)
Button(st,width=15,text='صورتحساب چاپ',command=print_bill).grid(row=5,column=6)
Button(st,width=15,text='صورتحساب ذخیره',command=make_bill).grid(row=7,column=6)

st.mainloop()

```

-----سیستم صورتحساب-----

نام را وارد کنید:

آدرس را وارد کنید:

آیدی (در صورت وجود)

---

انتخاب محصول	RACK	QTY	LEFT	COST	تعداد	ذخیره صورتحساب
1. combiflame	1	500		Rs 24		
2. disprin	3	414		Rs 12		
3. new	3	5		Rs 12		
4. alz142	2	116		Rs 850		
5. vomikind	6	548		Rs 45		
6. crocin	3	30		Rs 20		

اگر دکمه منوی اصلی را فشار دهیم مجدداً وارد صفحه منوی اصلی میشویم. با انتخاب دکمه صورتحساب وارد بخش مربوط به ایجاد صورتحساب میشویم که در آن 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 اضافه شده در این مرحله در دیتابیس ذخیره میشود.

ورودی انبار

داده محصول جدید انبار را وارد کنید

اسم:   
 نوع:   
 مقدار باقیمانده:   
 هزینه:   
 موثر برای:   
 تاریخ انقضا:   
 مکان قفسه:   
 ساخت:

را اندازی مجدد    ثبت    رفرش کردن انبار    منوی اصلی

اسم	نوع	مقدار باقیمانده	هزینه	موثر برای	exp	rack	manufacture
1. combiflame	aspirin	500	24	pain,fever	04-11-2017	1	medi
2. disprin	aspirin	414	12	headache	21-01-2022	3	medi
3. new	dn	5	12	dn	01-11-2017	3	old
4. alz142	tranquilizer	116	850	anti-depressant	05-11-2017	2	scime
5. vomikind	med	548	45	vomit	19-12-2017	6	mankind
6. crocin	aspirin	30	20	headache	10-02-2032	3	medi

```
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('انبار ورودی')
    Label(sto,text='وارد را انبار جدید محصول داده
    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='اندازی را
مجدد',command=reset).grid(row=12,column=0)
        Button(sto,width=15,text='کردن رفرش
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.



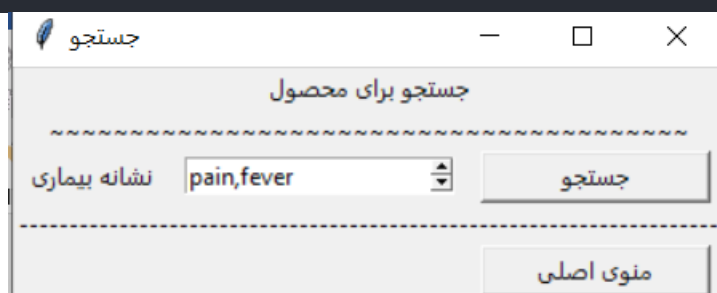
```
def delete_stock(): #OPENS DELETE WINDOW-----
-----DELETES A PARTICULAR STOCK ITEM
    global cur, c, flag, lbl, d
    apt.destroy()
    flag='d'
    d=Tk()
    d.title("حذف یک محصول از انبار")
    Label(d,text='وارد کنید برای حذف شدن محصول:').grid(row=0,column=0)
    Label(d,text='',width=30,bg='white').grid(row=0,column=1)
    Label(d,text='محصول').grid(row=2,column=0)
    Label(d,text='هزینه .انقضا تاریخ .تعداد').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()
```



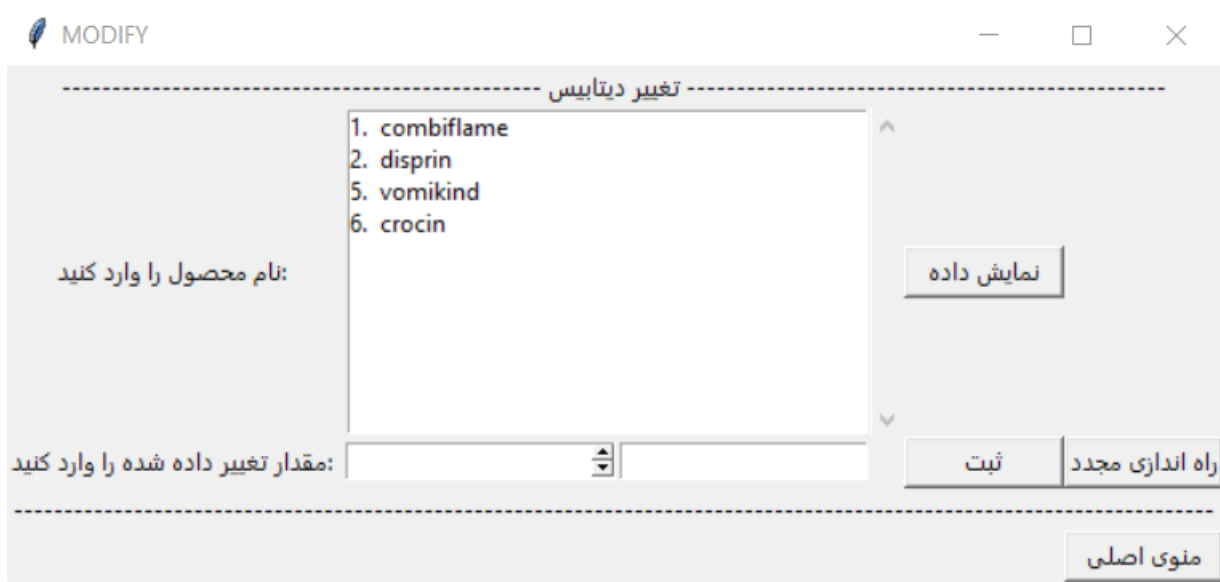


همچنین اگر روی دکمه حذف محصول از انبار در منوی اصلی کلیک کنیم، وارد صفحه بالا میشویم و میتوانیم یکی از محصولاتی که قبلاً در دیتابیس ذخیره شده و اکنون در یک متغیر ذخیره شده اند را انتخاب و آن را حذف کنیم تا از انبار حذف شود بلافاصله بعد از حذف کردن از لیست نیز حذف خواهد شد.

```
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='محصول برای جستجو ').grid(row=0,
column=0,columnspan=3)
    Label(st, text='~'*40).grid(row=1, column=0,columnspan=3)
    Label(st, text='بیماری نشانه').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 منوی',
command=main_cus).grid(row=6, column=2)
    else:
        Button(st,width=15, text='منوی منوی',
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 ذخیره 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+' دیتابیس تغییر '+'*48).grid(row=0,
column=0,columnspan=6)
    def onvsb(*args):
        name_.yview(*args)
    def onmousewheel():
        name_.yview=('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)
```

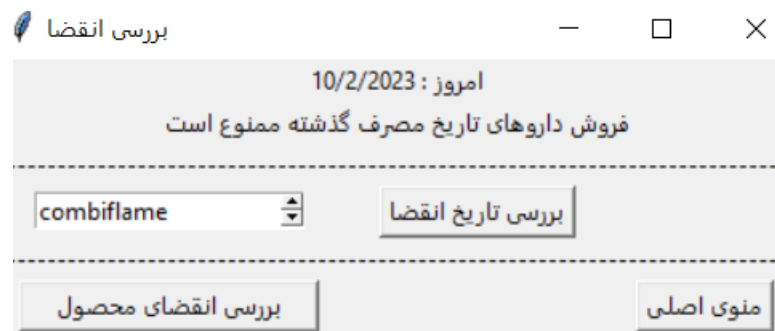
```

name_.bind('<<ListboxSelect>>', sel_mn)

Label(st, text='کنید وارد را محصول نام: ').grid(row=1, column=0)
Label(st, text='کنید وارد را شده داده تغییر مقدار: ').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='مجدد اندازی راه', command=res).grid(row=2,
column=5)
Button(st,width=10,text='داده نمایش', 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 محصولات را 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 ممنوع گذشته مصرف تاریخ داروهای فروش').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='انقضا تاریخ بررسی', 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='اصلی منوی', command=main_cus).grid(row=5,
column=2)
else:
    Button(exp, width=20, text='محصول انقضای بررسی',
command=exp_dt).grid(row=5, column=0)
    Button(exp, text='اصلی منوی', 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 درآمد 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 در یک جدول از دیتابیس ذخیره شده است.