Pytorn

Monday, April 22, 2019 11:56 AM

def membersignup():

```
root2 = Tk()
root2.title = "สมัครสมาชิก"
root2.geometry("200x300")
app2 = Frame(root2)
app2.grid()
Label1 = Label(app2, text="ชื่อผู้ใช้")
Label1.grid(column=0, row=0)
Entry1 = Entry(app2, bd=2)
Entry1.grid(column=2, row=0)
Label2 = Label(app2, text="รหัสผ่าน")
Label2.grid(column=0, row=2)
Entry2 = Entry(app2, bd=2)
Entry2.grid(column=2, row=2)
Label3 = Label(app2, text="ยืนยันรหัสผ่าน")
Label3.grid(column=0, row=4)
Entry3 = Entry(app2, bd=2)
Entry3.grid(column=2, row=4)
b2 = Button(app2, text="ยืนยันการสมัคร",
command=lambda: checksignup(Entry1.get(),
Entry2.get(), Entry3.get()))
b2.grid(column=1, row=8)
root2.mainloop()
```

def checksignup(data1, data2, data3):

```
if data2 == data3:
    insertsignup(data1, data2)
else:
    print("error")
```

def dropsignup() :

```
try:
    con = sqlite3.connect('Ex_163.db')
    con.execute('DROP TABLE IF EXISTS signup')
except con.Error as e:
    if con:
        print("error is " + e)
    con.close()
```

def createsignup():

```
try:
    con = sqlite3.connect('Ex_163.db')
    sql = 'CREATE TABLE signup ' \
          '(uname TEXT PRIMARY KEY NOT NULL, upass TEXT);'
    con.execute(sql)
except con.Error as e:
    if con:
        print("error is " + e)
    con.close()
```

def insertsignup(data1, data2):

```
dropsignup()
createsignup()

try:
    con = sqlite3.connect('Ex_163.db')
    sql = 'INSERT INTO signup ' \
          '(uname, upass) ' \
          'VALUES ("'+ data1 + '","' + data2 +'")'

    con.execute(sql)
    print(sql)
    con.commit()
except con.Error as e:
    if con:
        print("error is " + e)
    con.close()
```

def signup(uname, upass):

```
print(upass)
try:
    print("uname")
    con = sqlite3.connect('Ex_163.db')
    sql = 'SELECT * FROM signup WHERE uname ="' +
uname + '" and upass ="' + upass + '"'
    data = con.execute(sql)
    result = False
```

def createtable():

```
try:
    con = sqlite3.connect('productDB.db')
    sql = 'CREATE TABLE IF NOT EXISTS signup ' \
          '(uname TEXT PRIMARY KEY NOT NULL, upass TEXT);'
    con.execute(sql)
    con.commit()
    sql = 'CREATE TABLE IF NOT EXISTS product ' \
          '(pid INT PRIMARY KEY NOT NULL, ' \
          'pname TEXT, totalA INT, totalB INT);'
    con.execute(sql)
    con.commit()
    sql = 'CREATE TABLE IF NOT EXISTS transfer ' \
          '(tid TEXT PRIMARY KEY NOT NULL , ' \
          'pid TEXT, pqty INT);'
    con.execute(sql)
    con.commit()
except con.Error as e:
    if con:
        print("error is " + e)
con.close()
try:
    con = sqlite3.connect('quiz2.db')
    sql = 'SELECT * FROM login'
    data = con.execute(sql)
    print("%-8s%-10s" % ('ID', 'PASSWORD'))
    print('-------------------------------------------------------')
    for row in data:
        print("%-8s%-10s" % (row[0], row[1]))
    print('-------------------------------------------------------')
except con.Error as e:
    if con:
        print("error is " + e)
finally:
    if con:
        con.close()
```

def insertsignup(user, pw):

```
try:
    con = sqlite3.connect('productDB.db')
    sql = 'INSERT INTO signup ' \
          '(uname, upass) ' \
          'VALUES ("' + user + '", "'+ pw + '")'
    con.execute(sql)
    con.commit()
except con.Error as e:
    if con:
        print("error is " + e)
con.close()
```

def selectsignup(user, pw):

```
try:
    con = sqlite3.connect('productDB.db')
    sql = 'SELECT * FROM signup WHERE uname = "' + user + \
          '" AND upass = "' + pw +'"'
    data = con.execute(sql)
    result=False
    for row in data:
        result=str(row[0])+';'+str(row[1])
    return result
except con.Error as e:
    if con:
        print("error is " + e)
finally:
    if con:
        con.close()
```

def selectproduct():

```
try:
    con = sqlite3.connect('productDB.db')
    sql = 'SELECT * FROM product'
    data = con.execute(sql)
    result = ['รหัส:ชื่อ:สินค้าคลัง A:สินค้าคลัง B']
    for row in data:
result.append(row[0]+':'+row[1]+':'+str(row[2])+':'+str(row[3])
)
```

def selectforlogin(id, passw):

```
global result
try:
    con = sqlite3.connect('quiz2.db')
    sql = 'SELECT * FROM login WHERE uid ="' + id + '" and
upass ="' + passw + '"'
    data = con.execute(sql)
    result = False
    for row in data:
        result = row[0]+';'+row[1]
    return result
except con.Error as e:
    if con:
        print("error is " + e)
finally:
    if con:
        con.close()
```

def insertTable(pid, pname, totalA, totalB):

```
try:
    con = sqlite3.connect('quiz2.db')
    sql =  'INSERT INTO product (pid, pname, totalA, totalB) '\
           'VALUES ("' +
pid+'","'+pname+'",'+str(totalA)+','+str(totalB) + ')'
    print(sql)
    con.execute(sql)
    con.commit()
except con.Error as e:
    if con:
        print("error is "+e)
        con.rollback();
finally:
    if con:
        con.close()
```

def DeleteProduct(pid):

```
try:
    con = sqlite3.connect('quiz2.db')
    sql ='DELETE FROM product WHERE  pid = "'+ pid +'"'
    print(sql)
    con.execute(sql)
    con.commit()
except con.Error as e:
    if con:
        print("error is "+e)
        con.rollback();
finally:
    if con:
        con.close()
```

def selectTable():

```
try:
    con = sqlite3.connect('quiz2.db')
    sql = 'SELECT * FROM product'
    data = con.execute(sql)
    print("%-8s%-10s%5s%10s" % ('ID', 'NAME', 'totalA',
'totalB'))
    print('-------------------------------------------------------')
    for row in data:
        print("%-8s%-10s%5s%10s" % (row[0], row[1], row[2],
row[3]))
    print('-------------------------------------------------------')

    sql = 'SELECT transfer.*, product.pname FROM transfer,
product where transfer.pid = product.pid'
    data = con.execute(sql)
    print("%-8s%-10s%5s%10s" % ('TID', 'PID', 'PNAME',
'PQTY'))
    print('-------------------------------------------------------')
    for row in data:
        print("%-8s%-10s%5s%10s" % (row[0], row[1], row[3],
row[2]))
    print('-------------------------------------------------------')

except con.Error as e:
    if con:
```

```
        sql = 'SELECT * FROM signup WHERE uname ="' +
 uname + '" and upass ="' + upass + '"'
        data = con.execute(sql)
        result = False
        for row in data:
            result = row[0]+';'+row[1]

        if result == False:
            print("No ")
        else:
            print("pass")
 except con.Error as e:
        if con:
            print("error is " + e)
 finally:
        if con:
            con.close()
```

def memberlogin():

```
    root3 = Tk()
    root3.title = "ลงชื่อเข้าใช้"
    root3.geometry("200x300")
    app3= Frame(root3)
    app3.grid()
    Label1 = Label(app3, text="ชื่อผู้ใช้")
    Label1.grid(column=0, row=0)
    Entry4 = Entry(app3, bd=2)
    Entry4.grid(column=1, row=0)
    Label2 = Label(app3, text="รหัสผ่าน")
    Label2.grid(column=0, row=1)
    Entry5 = Entry(app3, bd=2)
    Entry5.grid(column=1, row=1)
    b2 = Button(app3, text="เข้าสู่ระบบ",
command=lambda:signup(Entry4.get(), Entry5.get()))
    b2.grid(column=0, row =2)
    root3.mainloop()
root = Tk()
root.title("Member Data")
root.geometry('250x300')
app = Frame(root)
app.grid()
b1 = Button(app, text="สมัครสมาชิก",
 command=membersignup)
b1.grid(column=1, row =8)
b2 = Button(app, text="เข้าสู่ระบบ", command=memberlogin)
b2.grid(column=1, row =10)

root.mainloop()



def selectTransferA():
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'SELECT transfer.*, product.pname FROM transfer, product where transfer.pid = product.pid and transfer.tid
like "%A"'
        print(sql)
        data = con.execute(sql)
        print("%-8s%-10s%5s%10s" % ('TID', 'PID', 'PNAME', 'PQTY'))
        print('-----------------------------------------------------------')
        for row in data:
            print("%-8s%-10s%5s%10s" % (row[0], row[1], row[3], row[2]))
        print('-----------------------------------------------------------')



    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()

def selectTransferB():
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'SELECT transfer.*, product.pname
FROM transfer, product where transfer.pid
 = product.pid and transfer.tid like "%B"'

        data = con.execute(sql)
        result = ['รหัส:ชื่อ:สินค้าคลัง A:สินค้าคลัง B']
        for row in data:
result.append(row[0]+':'+row[1]+':'+str(row[2])+':'+str(row[3])
)
        return result
    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()
```

def inserttransfertable(tid, id, tqty):

```
    try:
        con = sqlite3.connect('productDB.db')
        sql = 'INSERT INTO product ' \
              '(tid, pid, tqty) ' \
              'VALUES ("' + tid + '", "'+ id + '", "'+ tqty+ '")'
        con.execute(sql)
        con.commit()
    except con.Error as e:
        if con:
            print("error is " + e)
    con.close()
```

def selectuser():

```
    try:
        con = sqlite3.connect('productDB.db')
        sql = 'SELECT * FROM signup'
        data = con.execute(sql)
        result = ['ชื่อ:รหัสผ่าน']
        for row in data:
            result=str(row[0])+';'+str(row[1])

        return result
    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()
```

def insertproducttable(id, name, totalA, totalB):

```
    try:
        con = sqlite3.connect('productDB.db')
        sql = 'INSERT INTO product ' \
              '(id, name, totalA, totalB) ' \
              'VALUES ("' + id + '", "'+ name + '", "'+ totalA+ '", "'+
totalB+ '")'
        con.execute(sql)
        con.commit()
    except con.Error as e:
        if con:
            print("error is " + e)
    con.close()
```

def updateproducttable(tid, pid, pqty, qtyA, qtyB):

```
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'UPDATE product SET totalA =  ' + str(qtyA) + \
              ', totalB =  ' + str(qtyB) + \
              ' WHERE pid = "' + pid + '"'
        print(sql)
        con.execute(sql)
        con.commit()
        sql = 'INSERT INTO transfer ' \
              '(tid, pid, pqty) ' \
              'VALUES ("' + tid + '","' + pid + '",' + str(pqty) + ')'
        print(sql)
        con.execute(sql)
        con.commit()
    except con.Error as e:
        if con:
            print("error is " + e)
            con.rollback();
    finally:
        if con:
            con.close()
```

```
row[2]))
        print('-----------------------------------------------------------')

    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()
```

def selectfortransfer(id):

```
    global result
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'SELECT * FROM product WHERE pid ="' + id + '"'
        data = con.execute(sql)
        for row in data:
            result = str(row[0]) + ';' + str(row[1]) + ';' + str(row[2]) +
';' + str(row[3])
        return result
    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()
```

def manageTransfer(tid, pid, pqty, qtyA, qtyB):

```
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'UPDATE product SET totalA =  ' + str(qtyA) + \
              ', totalB =  ' + str(qtyB) + \
              ' WHERE pid = "' + pid + '"'
        print(sql)
        con.execute(sql)
        con.commit()
        sql = 'INSERT INTO transfer ' \
              '(tid, pid, pqty) ' \
              'VALUES ("' + tid + '","' + pid + '",' + str(pqty) + ')'
        print(sql)
        con.execute(sql)
        con.commit()
    except con.Error as e:
        if con:
            print("error is " + e)
            con.rollback();
    finally:
        if con:
            con.close()
```

def SelectGTransfer():

```
    try:
        con = sqlite3.connect('quiz2.db')
        sql = 'SELECT transfer.pid, product.pname,
sum(transfer.pqty) as total FROM transfer, product where
transfer.pid = product.pid group by transfer.pid'
        print(sql)
        data = con.execute(sql)

        print("%-8s%-10s%5s" % ('ID', 'NAME', 'SUM'))
        print('-----------------------------------------------------------')
        for row in data:
            print("%-8s%-10s%5s" % (row[0], row[1], row[2]))
        print('-----------------------------------------------------------')



    except con.Error as e:
        if con:
            print("error is " + e)
    finally:
        if con:
            con.close()
```

```
    print("%-8s%-10s%5s%10s" % ('TID', 'PID', 'PNAME', 'PQTY'))
    print('-------------------------------------------------------------')
    for row in data:
        print("%-8s%-10s%5s%10s" % (row[0], row[1], row[3], row[2]))
    print('-------------------------------------------------------------')


except con.Error as e:
    if con:
        print("error is " + e)
finally:
    if con:
        con.close()
```