







headTail.py

def headHtml():  
 print("Content\_type: 'text/html' \n\n")  
 print("<html>")  
 print("<head>")  
 print("</head>")  
 print("<body>")  
  
  
def tailHtml():  
 print("</body>")  
 print("</html>")

dbAccess.py

#!/Python27/python  
  
import cgi  
import sqlite3  
  
  
def findPassword(login):  
 db = "acct.db"  
 try:  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select password from pswdTbl  
 where login="{}" '''.format(login)  
 cursor.execute(sql)  
 dbPasswordList = cursor.fetchone()  
 if dbPasswordList == None:  
 dbPassword = None  
 else:  
 dbPassword = dbPasswordList[0]  
 conn.commit()  
 conn.close()  
 return dbPassword  
 except Exception as ex:  
 cgi.print\_exception()  
  
  
def findAcctIds(login):  
 db = "acct.db"  
 try:  
 acctIdList = []  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select id from acctTbl  
 where login="{}" '''.format(login)  
 cursor.execute(sql)  
 acctIdListSeq = cursor.fetchall()  
 for acctIdSeq in acctIdListSeq:  
 acctIdList.append(acctIdSeq[0])  
 conn.commit()  
 conn.close()  
 return acctIdList  
 except Exception as ex:  
 cgi.print\_exception()  
  
  
def findBalance(acctId):  
 db = "acct.db"  
 try:  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select balance from acctTbl  
 where id={}'''.format(acctId)  
 cursor.execute(sql)  
 balance = (cursor.fetchone())[0]  
 conn.commit()  
 conn.close()  
 return balance  
 except Exception as ex:  
 cgi.print\_exception()  
  
  
def deposit(acctId, amount):  
 db = "acct.db"  
 try:  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select balance from acctTbl  
 where id={}'''.format(acctId)  
 cursor.execute(sql)  
 balance = float((cursor.fetchone())[0])  
 newBalance = balance + amount  
 sql = '''update acctTbl set balance={}  
 where id={}'''.format(newBalance, acctId)  
 cursor.execute(sql)  
 conn.commit()  
 conn.close()  
 return newBalance  
 except Exception as ex:  
 cgi.print\_exception()  
  
  
def withdraw(acctId, amount):  
 db = "acct.db"  
 try:  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select balance from acctTbl  
 where id={}'''.format(acctId)  
 cursor.execute(sql)  
 balance = float((cursor.fetchone())[0])  
 newBalance = balance - amount  
 sql = '''update acctTbl set balance={}  
 where id={}'''.format(newBalance, acctId)  
 cursor.execute(sql)  
 conn.commit()  
 conn.close()  
 return newBalance  
 except Exception as ex:  
 cgi.print\_exception()  
  
  
def findAcctLogins(acctId):  
 db = "acct.db"  
 dbLogin = []  
 try:  
 conn = sqlite3.connect(db)  
 cursor = conn.cursor()  
 sql = '''select login from idLoginTbl  
 where id={}'''.format(acctId)  
 cursor.execute(sql)  
 dbLogin = cursor.fetchall()  
 conn.commit()  
 conn.close()  
 return dbLogin  
 except Exception as ex:  
 cgi.print\_exception()

formInput.py

#!/Python27/python  
  
import cgi  
#from email.policy import default  
  
def dispLoginForm():  
 print('''  
<form action="DoLogin.py" method="get">  
login:  
<input type="text" name="login" col="10" /><br>  
password:  
<input type="password" name="password" col="10" /><br>  
<input type="submit" name="submit" value="Submit" /><br>  
''')  
  
def getLoginInput():  
 f = cgi.FieldStorage()  
 login = f.getvalue('login', default='')  
 password = f.getvalue('password', default='')  
 return login, password  
  
  
def dispAcctListForm(login, acctList):  
 print('''  
<form action="AcctAccess.py">  
<br>login={}  
<br>  
<br>From the drop-down list of your accounts below,   
<br>select one account to work on  
<br>  
 <select name="acctSelected">  
'''.format(login))  
 print("<option value=></option>")  
 for acct in acctList:  
 print("<option value={}>{}</option>".format(acct, acct))  
 print('''  
 </select>  
 <br><br>  
 <input type = "submit" name = "submit"   
 value = "Submit" />  
</form>  
''')  
  
  
def getAcctListInput():  
 f = cgi.FieldStorage()  
 acctSelected = int(f.getvalue('acctSelected'))  
 return acctSelected  
  
  
def dispAcctForm(acctId, balance):  
 print('''  
<br><form action="DoAcctAccess.py" method="get">  
<br>Account Id: {}  
<input type="hidden" name="acctId" value="{}" />  
<br>Current Balance: {}  
<br>Enter Deposit/Withdrawal Amount:  
<br><input type="text" name="amount" col="10" autofocus />  
<br><br><input type="radio" name="op" value="deposit" checked />Deposit  
<input type="radio" name="op" value="withdraw" />Withdraw  
<br><br><input type="submit" name="submit" value="Submit" />  
<br><br><input type="submit" name="submit" value="Done" />  
<br></form><br>  
'''.format(acctId, acctId, balance))  
  
  
def getAcctInput():  
 f = cgi.FieldStorage()  
 acctId = int(f.getvalue('acctId', default=-1))  
 amount = float(f.getvalue('amount', default=0))  
 op = f.getvalue('op', default='')  
 submit = f.getvalue('submit', default='')  
 return acctId, amount, op, submit  
  
  
def dispSharedLogins(acctId, logins):  
 print('''  
 <br>acct id# acct sharers<br>  
 ''')  
 for login in logins:  
 convert = str(login)  
 print('''{}&emsp;&emsp;&emsp;{}<br>  
 '''.format(acctId, convert.split("\'")[1]))

Login.py

#!/Python27/python  
  
import cgi  
from headTail import headHtml  
from headTail import tailHtml  
from formInput import dispLoginForm  
  
def main():  
 headHtml()  
 dispLoginForm()  
 tailHtml()  
 return  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 main()  
 except:  
 cgi.print\_exception()

DoLogin.py

#!/Python27/python  
  
import cgi  
from headTail import headHtml  
from headTail import tailHtml  
from dbAccess import findPassword  
from dbAccess import findAcctIds  
from formInput import dispAcctListForm  
from formInput import getLoginInput  
  
def main():  
 headHtml()  
 login, password = getLoginInput()  
 #print('login:'+login)  
 #print('password:' + password)  
 if findPassword(login) == None or \  
 findPassword(login) != password:  
 print("Invalid Login and Password ")  
 else:  
 acctList = findAcctIds(login)  
 dispAcctListForm(login, acctList)  
 tailHtml()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 main()  
 except:  
 cgi.print\_exception()

AcctAccess.py

#!/Python27/python  
  
import cgi  
from headTail import headHtml  
from headTail import tailHtml  
from formInput import getAcctListInput  
from formInput import dispAcctForm  
from formInput import dispSharedLogins  
from dbAccess import findBalance  
from dbAccess import findAcctLogins  
  
def main():  
 id = getAcctListInput()  
 logins = findAcctLogins(id)  
 headHtml()  
 dispAcctForm(id, findBalance(id))  
 dispSharedLogins(id, logins)  
 tailHtml()  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 main()  
 except:  
 cgi.print\_exception()

DoAcctAccess.py

#!/Python27/python  
  
import cgi  
from headTail import headHtml  
from headTail import tailHtml  
from formInput import getAcctInput  
from formInput import dispAcctForm  
from formInput import dispSharedLogins  
from dbAccess import findBalance  
from dbAccess import deposit  
from dbAccess import withdraw  
from dbAccess import findAcctLogins  
  
  
def main():  
 acctId, amount, op, submit = getAcctInput()  
 logins = findAcctLogins(acctId)  
  
 if submit == 'Submit':  
 if op == 'deposit':  
 deposit(acctId, amount)  
 headHtml()  
 dispAcctForm(acctId, findBalance(acctId))  
 dispSharedLogins(acctId, logins)  
 tailHtml()  
  
 if op == 'withdraw':  
 withdraw(acctId, amount)  
 headHtml()  
 dispAcctForm(acctId, findBalance(acctId))  
 dispSharedLogins(acctId, logins)  
 tailHtml()  
 else:  
 headHtml()  
 print('<br>Bye<br>')  
 tailHtml()  
 return  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 main()  
 except:  
 cgi.print\_exception()