PYTHON CODE IBM_DB CONNECTIONS

```
from flask import Flask, render_template, request, redirect, url_for, session
import mysql.connector
import re
import ibm_db
import time,random
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=21fecfd8-47b7-4937-840d-
d791d0218660.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31864;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=vpl67398;PWD=c8CTODyXc
Pc9RJTp",'','')
app = Flask(__name__)
app.secret_key = 'your secret key'

# cnx = mysql.connector.connect(user='root', password='',
host='127.0.0.1',  database='ibm_db')
# cursor = cnx.cursor(buffered=True)
@app.route('/')

@app.route('/index', methods =['GET', 'POST'])
def index():
    msg = ''
    if request.method == 'POST' and 'email' in request.form and 'password' in
request.form:
        email = request.form['email']
        password = request.form['password']
        username = request.form['username']
        # sql = 'SELECT * FROM accounts WHERE email = %s',(email,)
        # stmt = ibm_db.exec_immediate(conn,sql)
        # dic = ibm_db.fetch_both(stmt)
        sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
        stmt2 = ibm_db.prepare(conn, sql2)
        ibm_db.bind_param(stmt2, 1, email)
        ibm_db.execute(stmt2)
        account = ibm_db.fetch_both(stmt2)
        # cursor.execute('SELECT * FROM accounts WHERE email = %s',(email,))
        # account = cursor.fetchone()
        if account:
            msg = 'Account already exists !'
            return "<p>Account already exists !</p>"
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
           return "<p>Invalid Email address !</p>"
        else:
```

```
            totalBalance = 0
            incomeBalance = 0
            expenseBalance = 0
            # t = int(time.time())
            # ran = random.randrange(t,10000000000)
            # sql = 'INSERT INTO USERS (USERNAME, EMAIL, PASSWORD,
TOTALBALANCE, INCOMEBALANCE, EXPENSEBALANCE) VALUES ( %s, %s, %s, %s, %s,
%s)',(username, email, password, totalBalance, incomeBalance, expenseBalance,)
            # cursor.execute('INSERT INTO accounts (username, email, password,
totalBalance, incomeBalance, expenseBalance) VALUES ( %s, %s, %s, %s, %s,
%s)',(username, email, password, totalBalance, incomeBalance,
expenseBalance,))
            # stmt = ibm_db.exec_immediate(conn,sql)
            # cnx.commit()

            #INSERT THE USER

            sql2 = "INSERT INTO USERS (USERNAME, EMAIL, PASSWORD,
TOTALBALANCE, INCOMEBALANCE, EXPENSEBALANCE) VALUES ( ?,?,?,?,?,?)"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, username)
            ibm_db.bind_param(stmt2, 2, email)
            ibm_db.bind_param(stmt2, 3, password)
            ibm_db.bind_param(stmt2, 4, totalBalance)
            ibm_db.bind_param(stmt2, 5, incomeBalance)
            ibm_db.bind_param(stmt2, 6, expenseBalance)
            result = ibm_db.execute(stmt2)
            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))
            # sql = 'SELECT * FROM accounts WHERE email = %s',(email,)
            # account = cursor.fetchone()

            # GET THE USER

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)
            name = account['USERNAME'].upper()
            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
            # session['loggedin'] = True
            # session['userID'] = account['userID']
            session['email'] = request.form.get("email")
            return render_template('profile.html',totalBalance=totalBalance,
incomeBalance=incomeBalance,expenseBalance=expenseBalance,len=0, name=name)
    else:
```

```
        return render_template('index.html')
@app.route('/login')
def login():
    return render_template('login.html')
@app.route('/login', methods =['GET', 'POST'])
def loginIn():
    msg = ''
    if request.method == 'POST' and 'email' in request.form and 'password' in
request.form:
        email = request.form['email']
        password = request.form['password']
        # sql = 'SELECT * FROM USERS WHERE EMAIL = %s AND PASSWORD =
%s',(email,password,)
        # cursor.execute('SELECT * FROM accounts WHERE email = %s AND password
= %s',(email,password,))
        # account = cursor.fetchone()

        #FETCH THE USER

        sql2 = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
        stmt2 = ibm_db.prepare(conn, sql2)
        ibm_db.bind_param(stmt2, 1, email)
        ibm_db.bind_param(stmt2, 2, password)
        ibm_db.execute(stmt2)
        account = ibm_db.fetch_both(stmt2)

        # stmt = ibm_db.exec_immediate(conn,sql)
        # dic = ibm_db.fetch_both(stmt)
        if account:
            # session['loggedin'] = True
            # session['userID'] = account['userID']
            session['email'] = request.form.get("email")
            email = session['email']
            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))

            #GET THE USER

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)

            # sql = 'SELECT * FROM accounts WHERE email = %s',(email,)
            # account = cursor.fetchone()
            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
```

```
            name = account['USERNAME'].upper()
            incomeBalance = account['INCOMEBALANCE']
            totalBalance = account['TOTALBALANCE']
            expenseBalance = account['EXPENSEBALANCE']

            sql = "SELECT * FROM HISTORY WHERE EMAIL = " + "\'" + email + "\'"
            stmt = ibm_db.exec_immediate(conn, sql)
            dictionary = ibm_db.fetch_assoc(stmt)
            l=[]
            print(dictionary)
            while dictionary != False:
                print(dictionary)
                l.append(list(dictionary.values()))
                dictionary = ibm_db.fetch_assoc(stmt)
            print(l[0][0])
            return render_template('profile.html',
totalBalance=totalBalance,incomeBalance=incomeBalance,
expenseBalance=abs(expenseBalance), len=len(l), l=l, name = name)
        else:
            return "<h1>Invalid</h1>"

@app.route('/add', methods =['POST','GET'])
def add():
    if request.method == 'POST' and 'title' in request.form and 'amount' in
request.form:
        title = request.form['title']
        amount = request.form['amount']
        email =  session["email"]
        if int(amount) > 0:

            #FETCH THE USER

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)
            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))
            # sql = 'SELECT * FROM accounts WHERE email = %s',(email,)
            # account = cursor.fetchone()
            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
            incomeBalance = account['INCOMEBALANCE']
            totalBalance = account['TOTALBALANCE']
            expenseBalance = account['EXPENSEBALANCE']
            incomeBalance += int(amount)
            totalBalance += int(amount)
```

```
            # sql = 'UPDATE USERS SET TOTALBALANCE = %s, INCOMEBALANCE = %s
WHERE EMAIL = %s',(totalBalance,incomeBalance,email,)
            # cursor.execute('UPDATE accounts SET totalBalance = %s,
incomeBalance = %s WHERE email = %s',(totalBalance,incomeBalance,email,))

            #UPDATE THE TRANSACTION

            sql2 = "UPDATE USERS SET TOTALBALANCE = ?, INCOMEBALANCE = ? WHERE
EMAIL = ?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, totalBalance)
            ibm_db.bind_param(stmt2, 2, incomeBalance)
            ibm_db.bind_param(stmt2, 3, email)
            ibm_db.execute(stmt2)


            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)

            #FETCH THE USER DETAILS

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)

            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))
            # account = cursor.fetchone()
            name = account['USERNAME'].upper()
            incomeBalance = account['INCOMEBALANCE']
            totalBalance = account['TOTALBALANCE']

            sql2 = "INSERT INTO HISTORY (EMAIL, TITLE, AMOUNT) VALUES (
?,?,?)"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.bind_param(stmt2, 2, title)
            ibm_db.bind_param(stmt2, 3, amount)
            result = ibm_db.execute(stmt2)



            sql = "SELECT * FROM HISTORY WHERE EMAIL = " + "\'" + email + "\'"
            stmt = ibm_db.exec_immediate(conn, sql)
            dictionary = ibm_db.fetch_assoc(stmt)
            l=[]
```

```
            print(dictionary)
            while dictionary != False:
                print(dictionary)
                l.append(list(dictionary.values()))
                dictionary = ibm_db.fetch_assoc(stmt)
            print(l[0][0])




            # cursor.execute('INSERT INTO history (email, title, amount )
VALUES ( %s, %s, %s)',(email, title, amount,))
            # cursor.execute('SELECT * FROM history WHERE email =
%s',(email,))
            # account = cursor.fetchall()

            # for i in account:
            #     print(i)
            return render_template('profile.html',
totalBalance=abs(totalBalance),incomeBalance=incomeBalance,expenseBalance=abs(
expenseBalance), len = len(l), name= name, l=l)
        elif int(amount) < 0:
            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))
            # sql ='SELECT * FROM USERS WHERE EMAIL = %s',(email,)
            # account = cursor.fetchone()

            #FETCH THE DETAILS

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)

            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
            incomeBalance =account['INCOMEBALANCE']
            totalBalance = account['TOTALBALANCE']
            print(totalBalance)
            expenseBalance = account['EXPENSEBALANCE']
            expenseBalance += int(amount)
            totalBalance -= abs(int(amount))
            print(totalBalance)

            #UPDATE THE DETAILS

            sql2 = "UPDATE USERS SET TOTALBALANCE = ?, EXPENSEBALANCE = ?
WHERE EMAIL = ?"
```

```
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, totalBalance)
            ibm_db.bind_param(stmt2, 2, expenseBalance)
            ibm_db.bind_param(stmt2, 3, email)
            ibm_db.execute(stmt2)

            # cursor.execute('UPDATE accounts SET totalBalance = %s,
expenseBalance = %s WHERE email = %s',(totalBalance,expenseBalance,email,))
            # sql = 'UPDATE USERS SET TOTALBALANCE = %s, EXPENSEBALANCE = %s
WHERE EMAIL = %s',(totalBalance,expenseBalance,email,)

            #FETCH THE DETAILS

            sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.execute(stmt2)
            account = ibm_db.fetch_both(stmt2)

            # cursor.execute('SELECT * FROM accounts WHERE email =
%s',(email,))
            # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
            # account = cursor.fetchone()
            name= account['USERNAME'].upper()
            expenseBalance = account['EXPENSEBALANCE']
            totalBalance = account['TOTALBALANCE']
            print(totalBalance)
            # cursor.execute('INSERT INTO history (email, title, amount )
VALUES ( %s, %s, %s)',(email, title, amount,))
            # cursor.execute('SELECT * FROM history WHERE email =
%s',(email,))
            # account = cursor.fetchall()
            # for i in account:
            #     print(i)


            sql2 = "INSERT INTO HISTORY (EMAIL, TITLE, AMOUNT) VALUES (
?,?,?)"
            stmt2 = ibm_db.prepare(conn, sql2)
            ibm_db.bind_param(stmt2, 1, email)
            ibm_db.bind_param(stmt2, 2, title)
            ibm_db.bind_param(stmt2, 3, amount)
            result = ibm_db.execute(stmt2)



            sql = "SELECT * FROM HISTORY WHERE EMAIL = " + "\'" + email + "\'"
```

```
            stmt = ibm_db.exec_immediate(conn, sql)
            dictionary = ibm_db.fetch_assoc(stmt)
            l=[]
            print(dictionary)
            while dictionary != False:
                print(dictionary)
                l.append(list(dictionary.values()))
                dictionary = ibm_db.fetch_assoc(stmt)
            print(l[0][0])
            return render_template('profile.html',
totalBalance=abs(totalBalance),incomeBalance=incomeBalance,expenseBalance=abs(
expenseBalance),len = len(l), l=l, name=name)

@app.route('/reset')
def reset():
    email =  session["email"]
    totalBalance = 0
    incomeBalance = 0
    expenseBalance = 0
    # cursor.execute('SELECT * FROM accounts WHERE email = %s',(email,))
    #         # stmt = ibm_db.exec_immediate(conn,sql)
            # dic = ibm_db.fetch_both(stmt)
    # account = cursor.fetchone()

    sql2 = "SELECT * FROM USERS WHERE EMAIL=?"
    stmt2 = ibm_db.prepare(conn, sql2)
    ibm_db.bind_param(stmt2, 1, email)
    ibm_db.execute(stmt2)
    account = ibm_db.fetch_both(stmt2)

    name=account['USERNAME']
    # cursor.execute('UPDATE accounts SET totalBalance = %s, expenseBalance =
%s, incomeBalance = %s WHERE email = %s',(0,0,0,email,))

    sql2 = "UPDATE USERS SET TOTALBALANCE = ?, INCOMEBALANCE = ? ,
EXPENSEBALANCE = ? WHERE EMAIL = ?"
    stmt2 = ibm_db.prepare(conn, sql2)
    ibm_db.bind_param(stmt2, 1, totalBalance)
    ibm_db.bind_param(stmt2, 2, incomeBalance)
    ibm_db.bind_param(stmt2, 3, expenseBalance)
    ibm_db.bind_param(stmt2, 4, email)
    ibm_db.execute(stmt2)
    # cursor.execute('DELETE FROM history WHERE email = %s',(email,))

    #HISTORY
    sql = "DELETE FROM HISTORY WHERE EMAIL = " + "\'" + email + "\'"
    stmt = ibm_db.exec_immediate(conn, sql)
```

```
    return render_template('profile.html',
totalBalance=0,incomeBalance=0,expenseBalance=0,len = 0,name=name)


@app.route('/logout')
def logout():
    # session.pop('loggedin', None)
    # session.pop('email', None)
    session["email"] = None
    # session.pop('userID', None)
    return redirect(url_for('login'))
```