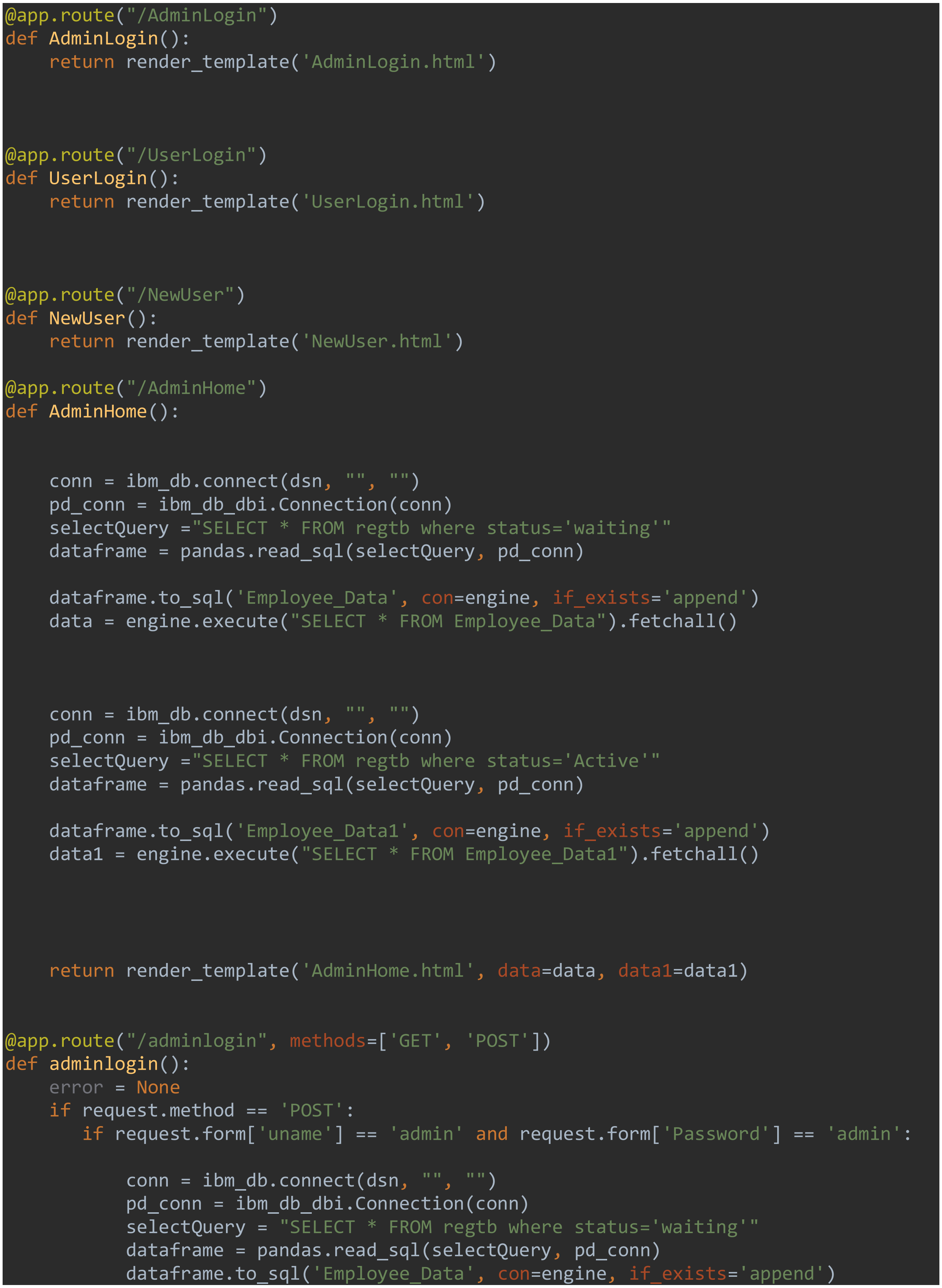
TEAM ID : PNT2022TMID30537

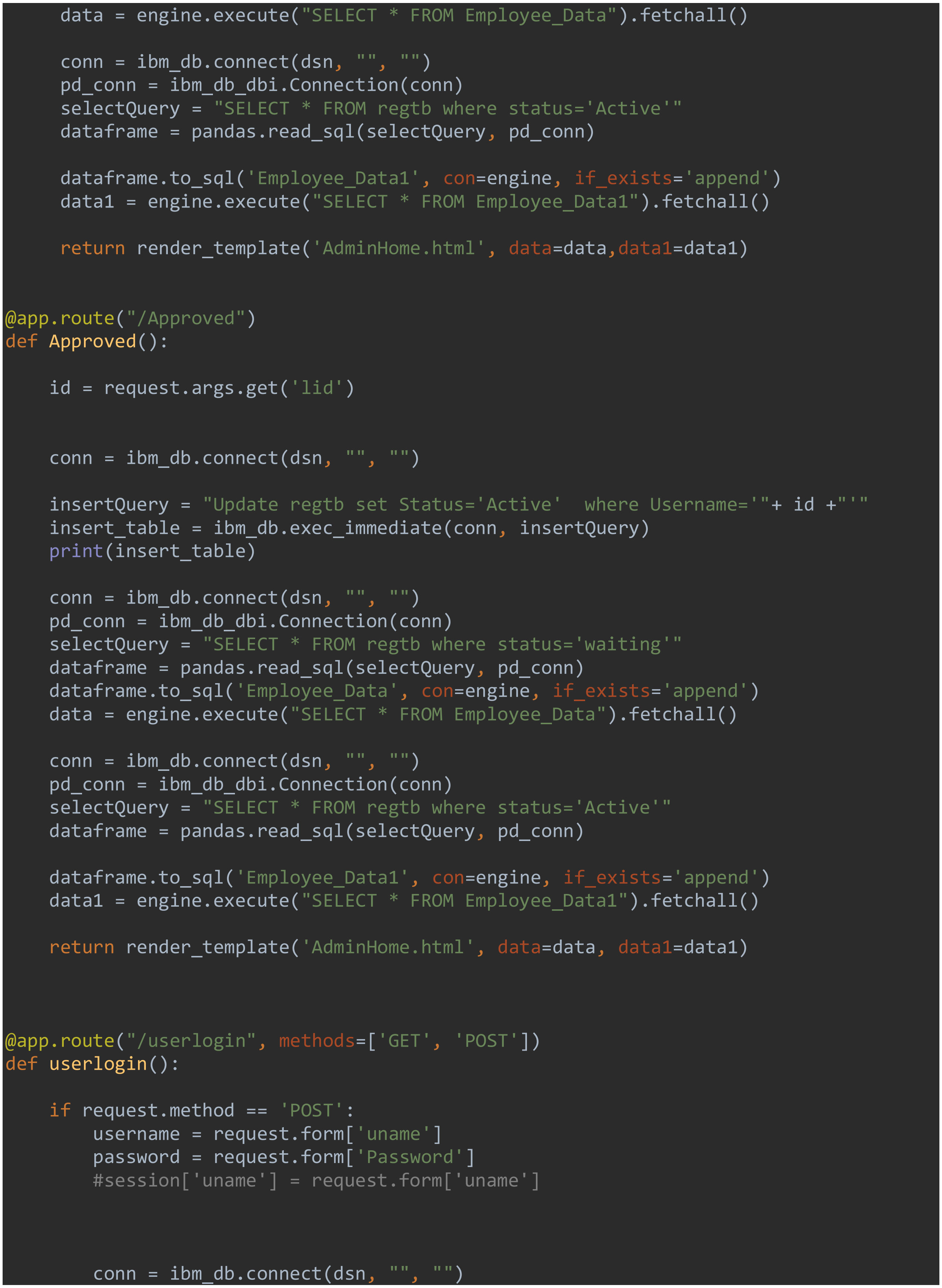
BUILD A PYTHON CODE:

```
import ibm_db
import pandas
import ibm_db_dbi
from sqlalchemy import create_engine

engine = create_engine('sqlite://',
                       echo = False)

dsn_hostname = "2f3279a5-73d1-4859-88f0-
a6c3e6b4b907.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud"
dsn_uid = "kfb92947"
dsn_pwd = "LllsyGThPcwKZPyv"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"
dsn_port = "30756"
dsn_protocol = "TCPIP"
dsn_security = "SSL"

dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port,
dsn_protocol, dsn_uid, dsn_pwd,dsn_security)



try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host:
", dsn_hostname)

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )

app = Flask(__name__)
app.config['DEBUG']
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'



@app.route("/")
def homepage():
    return render_template('index.html')
```

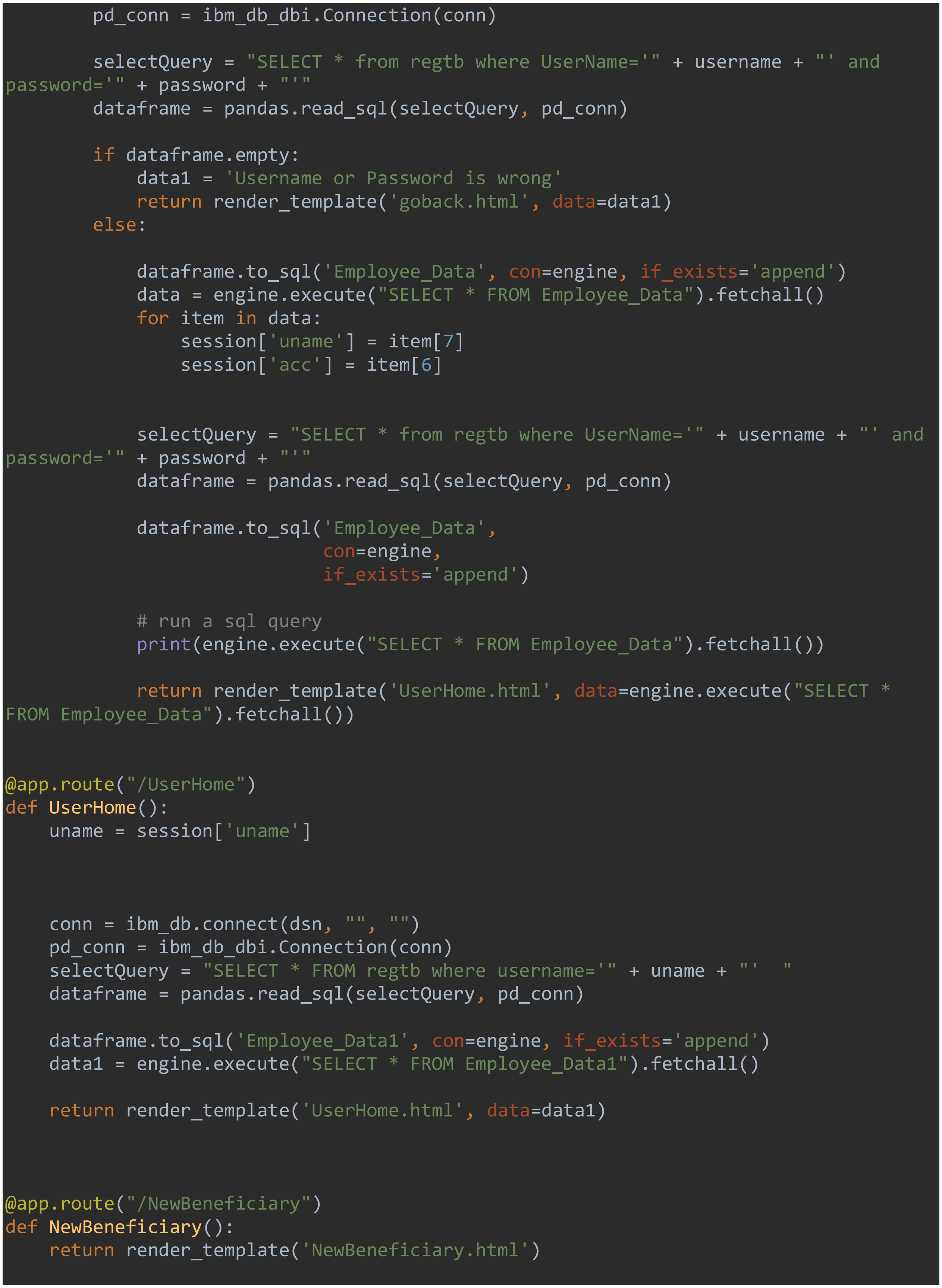


```
@app.route("/AdminLogin")
def AdminLogin():
    return render_template('AdminLogin.html')



@app.route("/UserLogin")
def UserLogin():
    return render_template('UserLogin.html')



@app.route("/NewUser")
def NewUser():
    return render_template('NewUser.html')

@app.route("/AdminHome")
def AdminHome():


    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery ="SELECT * FROM regtb where status='waiting'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()



    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery ="SELECT * FROM regtb where status='Active'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()




    return render_template('AdminHome.html', data=data, data1=data1)


@app.route("/adminlogin", methods=['GET', 'POST'])
def adminlogin():
    error = None
    if request.method == 'POST':
       if request.form['uname'] == 'admin' and request.form['Password'] == 'admin':

           conn = ibm_db.connect(dsn, "", "")
           pd_conn = ibm_db_dbi.Connection(conn)
           selectQuery = "SELECT * FROM regtb where status='waiting'"
           dataframe = pandas.read_sql(selectQuery, pd_conn)
           dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
```

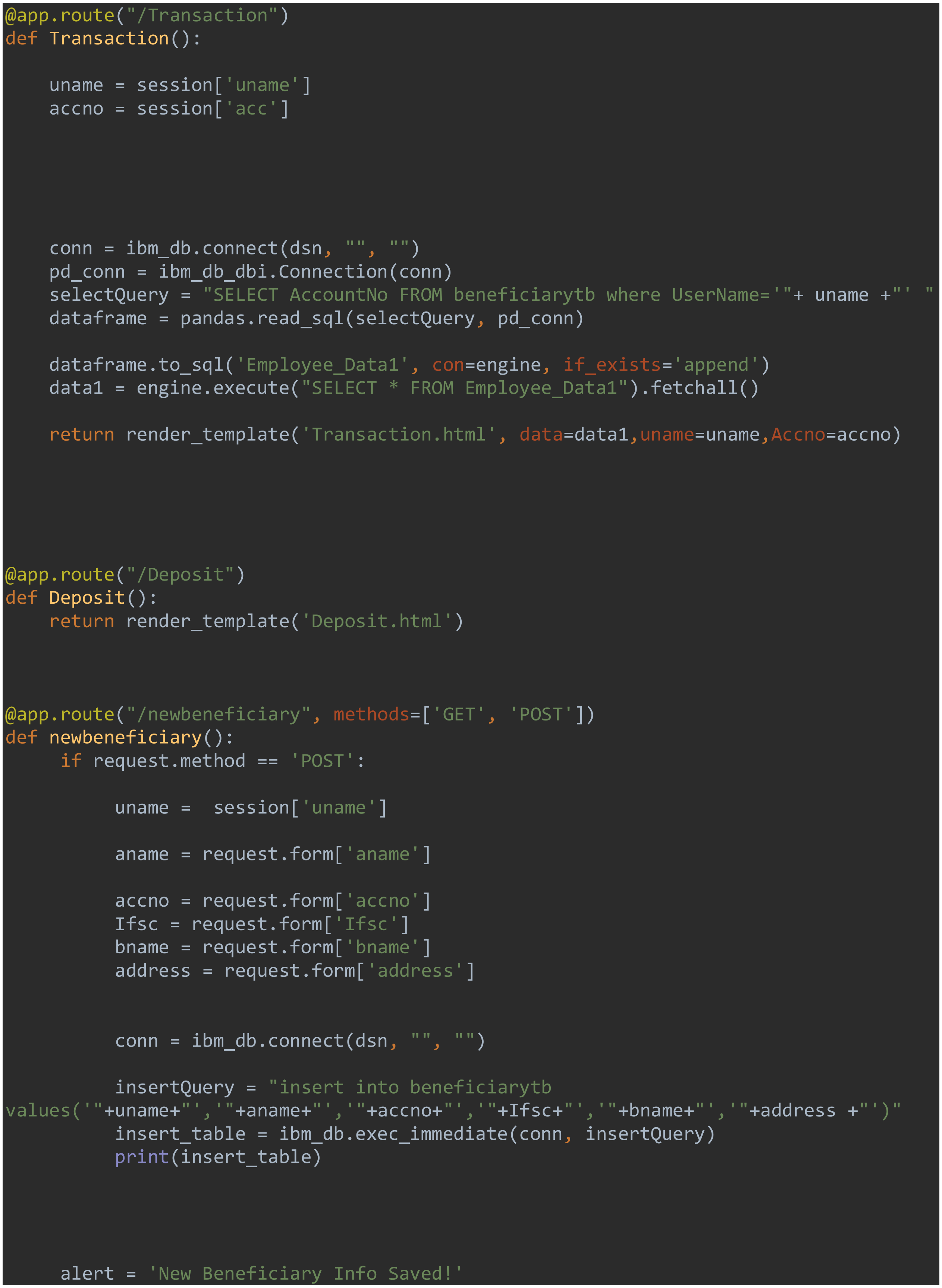
```
            data = engine.execute("SELECT * FROM Employee_Data").fetchall()




            conn = ibm_db.connect(dsn, "", "")
            pd_conn = ibm_db_dbi.Connection(conn)
            selectQuery = "SELECT * FROM regtb where status='Active'"
            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
            data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

            return render_template('AdminHome.html', data=data, data1=data1)

        else:
            data = "UserName or Password Incorrect!"

            return render_template('goback.html', data=data)






@app.route("/newuser", methods=['GET', 'POST'])
def newuser():
    if request.method == 'POST':

        name = request.form['name']

        age = request.form['age']
        mobile = request.form['mobile']
        email = request.form['email']
        address = request.form['address']
        accno = request.form['accno']
        username = request.form['username']
        Password = request.form['Password']

        conn = ibm_db.connect(dsn, "", "")

        insertQuery = "insert into regtb
values('"+name+"','"+age+"','"+mobile+"','"+email+"','"+address+"','"+accno
+"','"+username+"','"+Password+"','waiting','0.00')"
        insert_table = ibm_db.exec_immediate(conn, insertQuery)
        print(insert_table)




    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where status='waiting'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
```



```
     data = engine.execute("SELECT * FROM Employee_Data").fetchall()

     conn = ibm_db.connect(dsn, "", "")
     pd_conn = ibm_db_dbi.Connection(conn)
     selectQuery = "SELECT * FROM regtb where status='Active'"
     dataframe = pandas.read_sql(selectQuery, pd_conn)

     dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
     data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

     return render_template('AdminHome.html', data=data,data1=data1)


@app.route("/Approved")
def Approved():

    id = request.args.get('lid')


    conn = ibm_db.connect(dsn, "", "")

    insertQuery = "Update regtb set Status='Active'  where Username='"+ id +"'"
    insert_table = ibm_db.exec_immediate(conn, insertQuery)
    print(insert_table)

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where status='waiting'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where status='Active'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

    return render_template('AdminHome.html', data=data, data1=data1)



@app.route("/userlogin", methods=['GET', 'POST'])
def userlogin():

    if request.method == 'POST':
        username = request.form['uname']
        password = request.form['Password']
        #session['uname'] = request.form['uname']



        conn = ibm_db.connect(dsn, "", "")
```



```
        pd_conn = ibm_db_dbi.Connection(conn)

        selectQuery = "SELECT * from regtb where UserName='" + username + "' and
password='" + password + "'"
        dataframe = pandas.read_sql(selectQuery, pd_conn)

        if dataframe.empty:
            data1 = 'Username or Password is wrong'
            return render_template('goback.html', data=data1)
        else:

            dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
            data = engine.execute("SELECT * FROM Employee_Data").fetchall()
            for item in data:
                session['uname'] = item[7]
                session['acc'] = item[6]


            selectQuery = "SELECT * from regtb where UserName='" + username + "' and
password='" + password + "'"
            dataframe = pandas.read_sql(selectQuery, pd_conn)

            dataframe.to_sql('Employee_Data',
                             con=engine,
                             if_exists='append')

            # run a sql query
            print(engine.execute("SELECT * FROM Employee_Data").fetchall())

            return render_template('UserHome.html', data=engine.execute("SELECT *
FROM Employee_Data").fetchall())


@app.route("/UserHome")
def UserHome():
    uname = session['uname']



    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM regtb where username='" + uname + "'   "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

    return render_template('UserHome.html', data=data1)



@app.route("/NewBeneficiary")
def NewBeneficiary():
    return render_template('NewBeneficiary.html')
```



```
@app.route("/Transaction")
def Transaction():

    uname = session['uname']
    accno = session['acc']





    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT AccountNo FROM beneficiarytb where UserName='"+ uname +"' "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

    return render_template('Transaction.html', data=data1,uname=uname,Accno=accno)





@app.route("/Deposit")
def Deposit():
    return render_template('Deposit.html')



@app.route("/newbeneficiary", methods=['GET', 'POST'])
def newbeneficiary():
     if request.method == 'POST':

          uname =  session['uname']

          aname = request.form['aname']

          accno = request.form['accno']
          Ifsc = request.form['Ifsc']
          bname = request.form['bname']
          address = request.form['address']


          conn = ibm_db.connect(dsn, "", "")

          insertQuery = "insert into beneficiarytb
values('"+uname+"','"+aname+"','"+accno+"','"+Ifsc+"','"+bname+"','"+address +"')"
          insert_table = ibm_db.exec_immediate(conn, insertQuery)
          print(insert_table)




     alert = 'New Beneficiary Info Saved!'
     return render_template('goback.html', data=alert)



import random
import datetime



@app.route("/transaction", methods=['GET', 'POST'])
def transaction():
     if request.method == 'POST':

         uname = session['uname']
         accno = session['acc']

         bacc = request.form['bacc']

         currency = request.form['currency']

         tcc= float(currency)

         date = datetime.datetime.now().strftime('%Y-%b-%d')

         conn = ibm_db.connect(dsn, "", "")
         pd_conn = ibm_db_dbi.Connection(conn)
         selectQuery ="SELECT  *  FROM   regtb where  UserName='" + uname + "'"
         dataframe = pandas.read_sql(selectQuery, pd_conn)

         dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
         data = engine.execute("SELECT * FROM Employee_Data").fetchall()
         for item in data:

             bal = item[10]

             Amount = float(bal) - float(tcc)

             print(Amount)

     selectQuery1 = "SELECT  *  FROM    beneficiarytb where  AccountNo='" + bacc +
"'"
     dataframe = pandas.read_sql(selectQuery1, pd_conn)

     dataframe.to_sql('regtb', con=engine, if_exists='append')
     data1 = engine.execute("SELECT * FROM regtb").fetchall()

     for item1 in data1:
         bname = item1[2]
     if(Amount < 0):

         alert = 'Amount Transaction Failed Balance:' + str(Amount)

         return render_template('goback.html', data=alert)
     else:
         conn = ibm_db.connect(dsn, "", "")

         insertQuery = "INSERT INTO  transtb VALUES ('" + uname + "','" + accno +
"','" + bname + "','" + bacc + "','" + currency + "','" + date + "')"
         insert_table = ibm_db.exec_immediate(conn, insertQuery)
         print(insert_table)

         alert = 'Amount Transaction Successfully Balance:' + str(Amount)

         return render_template('goback.html', data=alert)




@app.route("/deposit", methods=['GET', 'POST'])
def deposit():
     if request.method == 'POST':


          uname =  session['uname']

          amt = request.form['amt']

     conn = ibm_db.connect(dsn, "", "")
     pd_conn = ibm_db_dbi.Connection(conn)
     selectQuery = "SELECT  *  FROM   regtb where  UserName='" + uname + "'"
     dataframe = pandas.read_sql(selectQuery, pd_conn)

     dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
     data = engine.execute("SELECT * FROM Employee_Data").fetchall()
     for item in data:
         bal = item[10]

         Amount = float(bal) + float(amt)

         print(Amount)


     conn = ibm_db.connect(dsn, "", "")

     insertQuery = "Update regtb set Balance='"+ str(Amount) +"'  where  UserName='"
+ uname + "'"
     insert_table = ibm_db.exec_immediate(conn, insertQuery)
     print(insert_table)
```

```
     alert = 'Amount Deposit Successfully Balance:'+ str(Amount)

     return render_template('goback.html', data=alert)



@app.route("/TransactionInfo")
def TransactionInfo():

    uname = session['uname']

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery ="SELECT * FROM beneficiarytb where UserName='"+uname +"'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)
    dataframe.to_sql('Employee_Data', con=engine, if_exists='append')
    data = engine.execute("SELECT * FROM Employee_Data").fetchall()

    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery = "SELECT * FROM transtb where UserName='"+ uname +"'"
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()



    return render_template('TransactionInfo.html', data=data, data1=data1)


@app.route("/ATransactionInfo")
def ATransactionInfo():




    conn = ibm_db.connect(dsn, "", "")
    pd_conn = ibm_db_dbi.Connection(conn)
    selectQuery ="SELECT * FROM transtb "
    dataframe = pandas.read_sql(selectQuery, pd_conn)

    dataframe.to_sql('Employee_Data1', con=engine, if_exists='append')
    data1 = engine.execute("SELECT * FROM Employee_Data1").fetchall()

    return render_template('ATransactionInfo.html',  data1=data1)
```

```
if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)
```