Sprint-3

|  |  |
| --- | --- |
| Team ID | PNT2022TMID52447 |
| Project Name | Smart Fashion Recommender System |

Source code:

from flask import Flask, render\_template, flash, request,session  
from flask import Flask, render\_template, request, jsonify  
import datetime  
import re  
  
  
  
import ibm\_db  
import pandas  
import ibm\_db\_dbi  
from sqlalchemy import create\_engine  
  
engine = create\_engine('sqlite://',  
 echo = False)  
  
dsn\_hostname = "fbd88901-ebdb-4a4f-a32e-9822b9fb237b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"  
dsn\_uid = "pnv79770"  
dsn\_pwd = "4C8CDWDfkqreIAYX"  
dsn\_driver = "{IBM DB2 ODBC DRIVER}"  
dsn\_database = "BLUDB"  
dsn\_port = "32731"  
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.from\_object(\_\_name\_\_)  
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("/NewUser")  
def NewUser():  
  
 return render\_template('NewUser.html')  
@app.route("/UserLogin")  
def UserLogin():  
  
 return render\_template('UserLogin.html')  
  
  
  
  
  
  
  
  
  
@app.route("/AdminHome")  
def AdminHome():  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 selectQuery = "SELECT \* from regtb "  
 dataframe = pandas.read\_sql(selectQuery, pd\_conn)  
  
 dataframe.to\_sql('Employee\_Data',  
 con=engine,  
 if\_exists='append')  
  
 # run a sql query  
 data = engine.execute("SELECT \* FROM Employee\_Data").fetchall()  
  
 return render\_template('AdminHome.html', data=data)  
  
  
@app.route("/NewProduct")  
def NewProduct():  
  
 return render\_template('NewProduct.html')  
  
@app.route("/ProductInfo")  
def ProductInfo():  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 selectQuery = "SELECT \* from protb "  
 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('ProductInfo.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
  
  
  
@app.route("/SalesInfo")  
def SalesInfo():  
  
 return render\_template('SalesInfo.html')  
  
  
  
  
  
@app.route("/Search")  
def Search():  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 selectQuery = "SELECT \* from protb "  
 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('ViewProduct.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
  
  
  
@app.route("/viewproduct", methods=['GET', 'POST'])  
def viewproduct():  
  
 searc = request.form['subcat']  
  
  
  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 selectQuery = "SELECT \* from protb where SubCategory like '%" + searc + "%' "  
 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('ViewProduct.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
  
  
  
  
@app.route("/RNewUser", methods=['GET', 'POST'])  
def RNewUser():  
 if request.method == 'POST':  
  
 name1 = request.form['name']  
 gender1 = request.form['gender']  
 Age = request.form['age']  
 email = request.form['email']  
 address = request.form['address']  
 pnumber = request.form['phone']  
 uname = request.form['uname']  
 password = request.form['psw']  
  
 conn = ibm\_db.connect(dsn, "", "")  
  
 insertQuery = "INSERT INTO regtb VALUES ('" + name1 + "','" + gender1 + "','" + Age + "','" + email + "','" + pnumber + "','" + address + "','" + uname + "','" + password + "')"  
 insert\_table = ibm\_db.exec\_immediate (conn, insertQuery)  
 print(insert\_table)  
  
  
  
  
  
  
 return render\_template('userlogin.html')  
  
@app.route("/RNewProduct", methods=['GET', 'POST'])  
def RNewProduct():  
 if request.method == 'POST':  
  
  
 file = request.files['fileupload']  
 file.save("static/upload/" + file.filename)  
  
  
 ProductId =request.form['pid']  
 Gender =request.form['gender']  
 Category =request.form['cat']  
 SubCategory=request.form['subcat']  
 ProductType=request.form['ptype']  
 Colour=request.form['color']  
 Usage=request.form['usage']  
 ProductTitle=request.form['ptitle']  
 price = request.form['price']  
 Image= file.filename  
 ImageURL="static/upload/" + file.filename  
  
  
  
 conn = ibm\_db.connect(dsn, "", "")  
  
 insertQuery = "INSERT INTO protb VALUES ('"+ ProductId +"','" + Gender + "','" + Category + "','" + SubCategory + "','" + ProductType + "','" + Colour + "','"+Usage +"','"+ProductTitle+"','"+ Image +"','"+ ImageURL +"','"+ price +"')"  
 insert\_table = ibm\_db.exec\_immediate(conn, insertQuery)  
  
 data1 = 'Record Saved!'  
 return render\_template('goback.html', data=data1)  
  
  
  
  
  
  
  
@app.route("/userlogin", methods=['GET', 'POST'])  
def userlogin():  
 error = None  
 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:  
 print("Login")  
 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("/adminlogin", methods=['GET', 'POST'])  
def adminlogin():  
 error = None  
 if request.method == 'POST':  
 username = request.form['uname']  
 password = request.form['password']  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
  
 if(username=="admin" and password=="admin"):  
 selectQuery = "SELECT \* from regtb "  
 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('AdminHome.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
 else:  
 data1 = 'Username or Password is wrong'  
 return render\_template('goback.html', data=data1)  
  
  
  
  
  
  
  
@app.route("/Remove", methods=['GET'])  
def Remove():  
  
  
 pid = request.args.get('id')  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 insertQuery = "Delete from protb where id='"+ pid +"'"  
 insert\_table = ibm\_db.exec\_immediate(conn, insertQuery)  
  
  
 selectQuery = "SELECT \* from protb "  
 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('ProductInfo.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
  
  
  
@app.route("/fullInfo")  
def fullInfo():  
 pid = request.args.get('pid')  
 session['pid'] = pid  
  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
  
 selectQuery = "SELECT \* FROM protb where ProductId='" + pid + "' "  
 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('ProductFullInfo.html', data=engine.execute("SELECT \* FROM Employee\_Data").fetchall())  
  
  
  
@app.route("/Book", methods=['GET', 'POST'])  
def Book():  
 if request.method == 'POST':  
  
  
 uname = session['uname']  
 pid = session['pid']  
  
 qty = request.form['qty']  
  
 ctype = request.form['ctype']  
 cardno = request.form['cardno']  
 cvno = request.form['cvno']  
  
  
 Bookingid = ''  
 ProductName =''  
 UserName= uname  
 Mobile=''  
 Email=''  
 Qty = qty  
 Amount=''  
  
  
 CardType = ctype  
 CardNo = cardno  
 CvNo = cvno  
 date = datetime.datetime.now().strftime('%d-%b-%Y')  
  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
 selectQuery = "SELECT \* FROM protb where ProductId='" + pid + "' "  
 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:  
 ProductName = item[8]  
 price = item[11]  
 print(price)  
 Amount = float(price) \* float(Qty)  
  
 print(Amount)  
  
 selectQuery1 ="SELECT \* FROM regtb where UserName='" + uname + "'"  
 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:  
 Mobile = item1[5]  
 Email = item1[4]  
  
  
  
  
 selectQuery = "SELECT \* FROM booktb"  
 dataframe = pandas.read\_sql(selectQuery, pd\_conn)  
  
 dataframe.to\_sql('booktb', con=engine, if\_exists='append')  
 data2 = engine.execute("SELECT \* FROM booktb").fetchall()  
 count = 0  
  
 for item in data2:  
 count+=1  
  
  
  
  
 Bookingid="BOOKID00" + str(count)  
  
  
 insertQuery = "INSERT INTO booktb VALUES ('" + Bookingid + "','"+ ProductName +"','" + price + "','" + uname + "','" + Mobile + "','" + Email + "','" + str(Qty) + "','" + str(Amount) + "','"+ str(CardType) +"','"+ str(CardNo) +"','"+ str(CvNo) +"','"+ str(date) +"')"  
 insert\_table = ibm\_db.exec\_immediate(conn, insertQuery)  
  
 sendmsg(Email,"order received delivery in one week ")  
  
  
  
  
  
  
  
 selectQuery = "SELECT \* FROM booktb where UserName= '" + uname + "' "  
 dataframe = pandas.read\_sql(selectQuery, pd\_conn)  
  
 dataframe.to\_sql('booktb1', con=engine, if\_exists='append')  
 data = engine.execute("SELECT \* FROM booktb1").fetchall()  
  
 return render\_template('UOrderInfo.html', data=data)  
  
  
  
def sendmsg(Mailid,message):  
 import smtplib  
 from email.mime.multipart import MIMEMultipart  
 from email.mime.text import MIMEText  
 from email.mime.base import MIMEBase  
 from email import encoders  
  
 fromaddr = "sampletest685@gmail.com"  
 toaddr = Mailid  
  
 # instance of MIMEMultipart  
 msg = MIMEMultipart()  
  
 # storing the senders email address  
 msg['From'] = fromaddr  
  
 # storing the receivers email address  
 msg['To'] = toaddr  
  
 # storing the subject  
 msg['Subject'] = "Alert"  
  
 # string to store the body of the mail  
 body = message  
  
 # attach the body with the msg instance  
 msg.attach(MIMEText(body, 'plain'))  
  
 # creates SMTP session  
 s = smtplib.SMTP('smtp.gmail.com', 587)  
  
 # start TLS for security  
 s.starttls()  
  
 # Authentication  
 s.login(fromaddr, "hneucvnontsuwgpj")  
  
 # Converts the Multipart msg into a string  
 text = msg.as\_string()  
  
 # sending the mail  
 s.sendmail(fromaddr, toaddr, text)  
  
 # terminating the session  
 s.quit()  
  
  
@app.route("/UOrderInfo")  
def UOrderInfo():  
  
 uname = session['uname']  
  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
 selectQuery = "SELECT \* FROM booktb where UserName= '" + uname + "' "  
 dataframe = pandas.read\_sql(selectQuery, pd\_conn)  
 dataframe.to\_sql('booktb1', con=engine, if\_exists='append')  
 data = engine.execute("SELECT \* FROM booktb1").fetchall()  
  
  
 return render\_template('UOrderInfo.html', data=data)  
  
  
@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('booktb1', con=engine, if\_exists='append')  
 data = engine.execute("SELECT \* FROM booktb1").fetchall()  
  
  
 return render\_template('UserHome.html', data=data)  
  
  
@app.route("/ASalesInfo")  
def ASalesInfo():  
 conn = ibm\_db.connect(dsn, "", "")  
 pd\_conn = ibm\_db\_dbi.Connection(conn)  
 selectQuery = "SELECT \* FROM booktb "  
 dataframe = pandas.read\_sql(selectQuery, pd\_conn)  
 dataframe.to\_sql('booktb', con=engine, if\_exists='append')  
 data = engine.execute("SELECT \* FROM booktb").fetchall()  
  
 return render\_template('ASalesInfo.html', data=data)  
  
def main():  
 app.run(debug=True, use\_reloader=True)  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()