*2017029570 남혜빈*

@app.route("/")

def index():

return render\_template("index.html",session=session)

-사이트에 접속하면 바로 index.html을 실행시켜 로그인 정보를 입력받는다.

@app.route('/login',methods=['GET','POST'])

def login():

sid = request.form.get('sid')

passwd = request.form.get('passwd')

id = sid.split('@')

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

sql = f"SELECT T.tablename, C.local, C.domain, C.passwd, C.name FROM customer C, (SELECT tablename FROM pg\_tables WHERE schemaname='public' AND tablename='customer') T WHERE local='{id[0]}' AND domain='{id[1]}' AND passwd='{passwd}' UNION SELECT T.tablename, S.local, S.domain, S.passwd, S.name FROM seller S,(SELECT tablename FROM pg\_tables WHERE schemaname='public' AND tablename='seller') T WHERE local='{id[0]}' AND domain='{id[1]}' AND passwd='{passwd}' UNION SELECT T.tablename, D.local, D.domain, D.passwd, D.name FROM delivery D,(SELECT tablename FROM pg\_tables WHERE schemaname='public' AND tablename='delivery') T WHERE local='{id[0]}' AND domain='{id[1]}' AND passwd='{passwd}'"

cur.execute(sql)

rows = cur.fetchall()

if(len(rows)==0):

conn.close()

return render\_template('index.html',session=session, msg = False)

elif(len(rows) > 1):

session['sid'] = sid

session['name'] = rows[0]['name']

session['passwd']=passwd

for i in range(len(rows)):

print(rows[i]['tablename'])

return render\_template('usertype.html', option=[rows[x]['tablename'] for x in range(len(rows))])

conn.close()

else:

if (rows[0]['passwd'].strip()==passwd):

session['sid'] = sid

session['name'] = rows[0]['name']

session['passwd'] = passwd

session['type'] = rows[0]['tablename']

if session['type'] == 'customer':

key = list(contacts.keys())

if len(key) == 0:

contacts[f"{session['name']}".strip()] = []

else:

if key.count(f"{session['name']}".strip())==0:

contacts[f"{session['name']}".strip()] = []

key = list(bucket.keys())

if len(key) == 0:

bucket[f'{session["name"]}'.strip()]=[]

else:

if key.count(f"{session['name']}".strip())==0:

bucket[f"{session['name']}".strip()] = []

conn.close()

return redirect("/"+rows[0]['tablename'])

-로그인 정보를 입력 받으면 입력 받은 id를 @기준으로 나눠서 delivery, customer, seller, 테이블에서 local. Domain, passwd를 비교한다. 만약 있으면 로그인 성공이고 없으면 실패다.

로그인 타입이 여러 개면 usertype.html으로 가서 타입을 선택하게 한다.

@app.route("/logout")

def logout():

session = {'sid' : None, 'name' : None, 'type' : None, 'passwd' : None}

return render\_template("index.html",session=session)

-사용자가 로그아웃 키를 누르면 index.html로 가서 다시 로그인 정보를 받음.

@app.route("/seller")

def seller():

return render\_template("seller.html",session=session, msg=msg)

-사용자가 seller로 로그인하면 seller.html로 이동

@app.route("/customer")

def customer():

return render\_template("customer.html", session=session, msg=msg)

-사용자가 customer로 로그인하면 customer.html로 이동

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

def changeinfo():

if request.method == "GET":

return render\_template("changeinfo.html", msg=None)

elif request.method =="POST":

passwd = request.form.get('passwd')

name = request.form.get('name')

if passwd == "":

passwd = session['passwd']

if name == "":

name = session['name']

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

ch = False

if(passwd != session['passwd']):

sql = f"UPDATE {session['type']} SET passwd='{passwd}' WHERE local='{session['sid'].split('@')[0]}' AND domain='{session['sid'].split('@')[1]}' AND passwd='{session['passwd']}';"

session['passwd'] = passwd

cur.execute(sql)

ch = True

if(name != session['name']):

sql = f"UPDATE {session['type']} SET name='{name}' WHERE local='{session['sid'].split('@')[0]}' AND domain='{session['sid'].split('@')[1]}' AND passwd='{session['passwd']}';"

session['name']=name

cur.execute(sql)

cur.execute(f"select \* from {session['type']} where name='{name}'" )

rows = cur.fetchall()

ch = True

if ch==False:

return render\_template("changeinfo.html",msg=False)

else:

conn.commit()

conn.close()

return render\_template(session['type']+".html",session=session, msg=True)

-정보변경 키를 누르면 변경할 이름과 비밀번호를 입력 받고 update를 통해서 정보를 수정한다.

@app.route("/contact")

def contact():

check = False

if contacts[f"{session['name']}".strip()] == []:

return render\_template("contact.html", check=False, session=session)

else:

return render\_template("contact.html",check=True, contacts=contacts[f"{session['name']}".strip()], session=session)

-주소록 목록 키를 누르면 contacts 딕셔너리에 key가 사용자 이름인 value를 contact.html에 넘겨줘서 화면에 출력

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

def add():

if request.method == 'GET':

return render\_template("add.html",msg=None)

elif request.method == 'POST':

number = request.form.get('number')

if contacts[f"{session['name']}".strip()].count(number) != 0:

return render\_template("add.html",msg=False )

contacts[f"{session['name']}".strip()].append(number)

return render\_template("contact.html",check=True, msg="추가",contacts=contacts[f"{session['name']}".strip()],session=session)

-추가할 연락처를 입력받고 contacts 딕셔너리에 key가 사용자 이름인 value에 연락처 추가

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

def delete():

if request.method == 'GET':

return render\_template("delete.html", msg=None)

elif request.method == 'POST':

number = request.form.get('number')

if contacts[f"{session['name']}".strip()].count(number) == 0:

return render\_template("delete.html",msg=False)

contacts[f"{session['name']}".strip()].remove(number)

if len(contacts[f"{session['name']}".strip()]) == 0:

return render\_template("contact.html",check=False, msg="삭제", session=session)

else:

return render\_template("contact.html",check=True, msg="삭제",contacts=contacts[f"{session['name']}".strip()], session=session)

-삭제할 연락처를 입력 받고 contacts딕셔너리에서 key가 사용자 이름인 value리스트에서 찾아서 삭제

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

def change():

if request.method == 'GET':

return render\_template("change.html",msg=None)

elif request.method == 'POST':

old = request.form.get('old')

new = request.form.get('new')

if contacts[f"{session['name']}".strip()].count(old) == 0:

return render\_template("change.html",msg=False)

contacts[f"{session['name']}".strip()][contacts[f"{session['name']}".strip()].index(old)] = new

return render\_template("contact.html",check=True, msg="변경",contacts=contacts[f"{session['name']}".strip()],session=session)

-기존 연락처와 변경할 연락처를 입력 받고 contacts 딕셔너리에서 key가 사용자 이름인 value에서 값을 찾아 변경

@app.route("/store")

def store():

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

sql = f"SELECT store.sid, store.sname, menu.menu FROM store, menu, seller WHERE store.sid=menu.sid AND seller.sid=store.seller\_id AND seller.name='{session['name'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

return render\_template("store.html", store=rows)

-판매자가 가게목록을 보고자 하면 판매자의 가게를 select해서 /store 화면에 출력

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

def menuadd():

if request.method == 'GET':

return render\_template("menuadd.html",msg=None)

elif request.method == 'POST':

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

st = request.form.get('store')

new = request.form.get('menu')

sql = f"INSERT INTO menu VALUES ('{new}',{st})"

cur.execute(sql)

conn.commit()

sql = f"SELECT store.sid, store.sname, menu.menu FROM store, menu, seller WHERE store.sid=menu.sid AND seller.sid=store.seller\_id AND seller.name='{session['name'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

return render\_template("store.html", msg="추가", store=rows)

-추가할 메뉴와 가게를 입력 받아서 insert로 menu테이블에 추가한다

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

def menudelete():

if request.method == 'GET':

return render\_template("menudelete.html", msg=None)

elif request.method == 'POST':

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

st = request.form.get('store')

menu = request.form.get('menu')

sql = f"SELECT count(\*) FROM menu WHERE sid={st} AND menu='{menu}'"

cur.execute(sql)

rows = cur.fetchall()

if(rows[0]==[0]):

return render\_template("menudelete.html",msg=False)

else:

sql = f"DELETE FROM menu WHERE menu='{menu}' AND sid={st}"

cur.execute(sql)

conn.commit()

sql = f"SELECT store.sid, store.sname, menu.menu FROM store, menu, seller WHERE store.sid=menu.sid AND seller.sid=store.seller\_id AND seller.name='{session['name'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

return render\_template("store.html",msg="삭제",store=rows)

-삭제할 메뉴와 가게를 입력 받고 delete를 통해 menu테이블에서 메뉴 삭제

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

def menuchange():

if request.method == 'GET':

return render\_template("menuchange.html", msg=None)

elif request.method == 'POST':

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

st = request.form.get('store')

old = request.form.get('old')

new = request.form.get('new')

sql = f"SELECT count(\*) FROM menu WHERE sid={st} AND menu='{old}'"

cur.execute(sql)

rows = cur.fetchall()

if(rows[0]==[0]):

return render\_template("menuchange.html",msg=False)

else:

sql = f"UPDATE menu SET menu='{new}' WHERE menu='{old}' AND sid={st}"

cur.execute(sql)

conn.commit()

sql = f"SELECT store.sid, store.sname, menu.menu FROM store, menu, seller WHERE store.sid=menu.sid AND seller.sid=store.seller\_id AND seller.name='{session['name'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

return render\_template("store.html",msg="변경",store=rows)

-기존 메뉴와 변경할 메뉴, 가게를 입력 받아서 update를 통해 menu테이블의 정보를 수정한다.

@app.route("/pay")

def pay():

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

return render\_template("pay.html", pay=new)

-customer 테이블에서 payments를 select해서 이를 리스트타입으로 바꾸고 pay.html으로 전달해서 화면에 출력

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

def payadd():

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

if request.method=='GET':

return render\_template("payadd.html", msg=None)

else:

paytype=request.form.get('paytype')

if paytype=='카드':

cardnum = request.form.get('cardnum')

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

dic = {"type":"card", "data":{"card\_num": int(cardnum)}}

new.append(dic)

up = json.dumps(new)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

elif paytype=='계좌':

bankname = request.form.get('bankname')

acc=request.form.get('account')

cur.execute(f"select bid from bank where name='{bankname}'")

n=cur.fetchall()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

dic = {"type":"bank", "data":{"bid": int(n[0]['bid']), "acc\_num":int(acc)}}

new.append(dic)

up = json.dumps(new)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

return render\_template("pay.html",msg="추가", pay=new)

-추가할 결제수단을 입력 받아 pay딕셔너리에 추가

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

def paychange():

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

if request.method=='GET':

return render\_template("paychange.html", msg=None)

else:

paytype=request.form.get('paytype')

if paytype=='카드':

old = request.form.get('old')

new = request.form.get('new')

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

yes = json.loads(rows[0]['payments'])

for i in range(len(yes)):

if yes[i]['type']=='card':

if yes[i]['data']['card\_num']==int(old):

yes[i]['data']['card\_num']=int(new)

break

if i == len(yes)-1:

return render\_template("paychange.html", msg=False)

up = json.dumps(yes)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

elif paytype=='계좌':

bankname = request.form.get('bankname')

old=request.form.get('bold')

new=request.form.get('bnew')

cur.execute(f"select bid from bank where name='{bankname}'")

n=cur.fetchall()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

yes = json.loads(rows[0]['payments'])

for i in range(len(yes)):

if yes[i]['type']=='bank':

if yes[i]['data']['bid']==int(n[0]['bid']):

if yes[i]['data']['acc\_num']==int(old):

yes[i]['data']['acc\_num']=int(new)

break

if i == len(yes)-1:

return render\_template("paychange.html", msg=False)

up = json.dumps(yes)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

return render\_template("pay.html",msg="변경", pay=new)

-결제 수단 타입을 선택하고 기존 정보와 변경할 정보를 입력 받은 후 pay딕셔너리에서 정보 수정

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

def paydelete():

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

if request.method=='GET':

return render\_template("paydelete.html", msg=None)

else:

paytype=request.form.get('paytype')

if paytype=='카드':

num = request.form.get('cardnum')

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

yes = json.loads(rows[0]['payments'])

for i in range(len(yes)):

if yes[i]['type']=='card':

if yes[i]['data']['card\_num']==int(num):

del yes[i]

break

if i == len(yes)-1:

return render\_template("paydelete.html", msg=False)

up = json.dumps(yes)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

elif paytype=='계좌':

bankname = request.form.get('bankname')

num=request.form.get('account')

cur.execute(f"select bid from bank where name='{bankname}'")

n=cur.fetchall()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

yes = json.loads(rows[0]['payments'])

for i in range(len(yes)):

if yes[i]['type']=='bank':

if yes[i]['data']['bid']==int(n[0]['bid']):

if yes[i]['data']['acc\_num']==int(num):

del yes[i]

break

if i == len(yes)-1:

return render\_template("paydelete.html", msg=False)

up = json.dumps(yes)

cur.execute(f"update customer set payments='{up}' where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'")

conn.commit()

sql = f"select payments from customer where local='{session['sid'].split('@')[0].strip()}' and domain='{session['sid'].split('@')[1].strip()}' and passwd='{session['passwd'].strip()}'"

cur.execute(sql)

rows = cur.fetchall()

new = json.loads(rows[0]['payments'])

return render\_template("pay.html",msg="삭제", pay=new)

-결제 수단 타입을 선택하고 삭제할 정보를 입력한 후 pay딕셔너리에서 정보 삭제

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

def search():

menu = {}

if request.method=='GET':

return render\_template("order.html", msg=None)

else:

sname=request.form.get('name')

tag = request.form.get('tag')

address = request.form.get('address')

if sname!="":

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

cur.execute(f"select count(\*) from store where sname='{sname}'")

num = cur.fetchall()

if num[0] == [0]:

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

menu[near[i]['sid']] = cur.fetchall()

return render\_template("order.html", msg=False, near=near, menu=menu)

else:

cur.execute(f"select \* from store where sname='{sname}'")

row = cur.fetchall()

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

menu[near[i]['sid']] = cur.fetchall()

return render\_template("order.html", msg=True, store=row, menu=menu, near=near)

elif address!="":

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

cur.execute(f"select count(\*) from store where address='{address}'")

num = cur.fetchall()

if num[0] == [0]:

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

menu[near[i]['sid']] = cur.fetchall()

return render\_template("order.html", msg=False, near=near, menu=menu)

else:

cur.execute(f"select \* from store where sname='{address}'")

row = cur.fetchall()

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

menu[near[i]['sid']] = cur.fetchall()

return render\_template("order.html", msg=True, store=row, near=near, menu=menu)

-/order창에서 검색할 정보를 입력 받으면 정보에 맞는 column을 store테이블에서 select한 후 출력

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

def order():

menu = {}

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

n = near[i]['schedules'].replace('\"\"', '"')

time = json.loads(n)

for i in range(len(near)):

menu[near[i]['sid']] = []

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

rows=cur.fetchall()

for j in range(len(rows)):

menu[near[i]['sid']].append(rows[j]['menu'].strip())

if request.method == 'GET' :

return render\_template("order.html", msg=None, near=near, menu=menu, bucket=bucket[f'{session["name"]}'])

else:

st = request.form.get('near')

me = request.form.get(f'{st}')

cnt = request.form.get('cnt')

dic = {'store' : f'{st}', 'menu':f'{me}', 'cnt': cnt}

bucket[f'{session["name"]}'.strip()].append(dic)

return render\_template("order.html", bucket=bucket[f'{session["name"]}'.strip()], add=True, near=near, menu=menu)

-장바구니에 담을 가게, 메뉴, 수량을 입력 받고 bucket딕셔너리에 추가

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

def doorder():

menu = {}

conn = pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

sql = f"select store.sid, store.sname, store.schedules from (select \* from customer where name='{session['name']}') c, store order by sqrt(power(c.lat - store.lat, 2)+power(c.lng-store.lng,2))"

cur.execute(sql)

near=cur.fetchall()

for i in range(len(near)):

n = near[i]['schedules'].replace('\"\"', '"')

time = json.loads(n)

for i in range(len(near)):

menu[near[i]['sid']] = []

sql = f"select menu from menu where sid = {near[i]['sid']}"

cur.execute(sql)

rows=cur.fetchall()

for j in range(len(rows)):

menu[near[i]['sid']].append(rows[j]['menu'].strip())

sql = f"select payments from customer where name='{session['name'].strip()}'"

cur.execute(sql)

row = cur.fetchall()

pay = json.loads(row[0]['payments'])

if request.method=='GET':

if bucket[f'{session["name"]}'.strip()] == []:

return render\_template("order.html", menu=menu, bucket=[], near=near)

elif pay == []:

return render\_template("order.html", menu=menu, bucket=bucket[f'{session["name"]}'.strip()], near=near, pay=False)

else:

return render\_template("doorder.html", pay=pay)

else:

nows = datetime.now()

pays=request.form.get('payment')

lists=[]

for i in range(len(bucket[f'{session["name"]}'.strip()])):

lists.append(bucket[f'{session["name"]}'.strip()][i])

lists[i]['name']=f'{session["name"]}'.strip()

lists[i]['payments']=pays

lists[i]['time']=str(nows)

orderlist[nows] = lists

bucket[f'{session["name"]}'.strip()]=[]

return render\_template("customer.html",order=True, session=session)

-/doorder창에서 결제 수단을 선택하게 하고 bucket[사용자이름]의 정보와 결제 수단 정보, 사용자 이름을 orderlist에 주문한 시간을 key로 한 후 저장

@app.route("/orderprint")

def orderprint():

if orderlist=={}:

return render\_template("customer.html", order=False)

newlist=[]

for key in list(orderlist.keys()):

for i in range(len(orderlist[key])):

if orderlist[key][i]['name']==f'{session["name"]}'.strip():

newlist.append(orderlist[key][i])

return render\_template("orderprint.html", orderlist=newlist)

-orderlist에서 key가 사용자 이름인 value를 /orderprint 화면에 출력

@app.route("/delivery")

def delivery():

conn=pg.connect(conn\_str)

cur = conn.cursor(cursor\_factory=psycopg2.extras.DictCursor)

lists=orderlist

for key in list(orderlist.keys()):

for i in range(len(orderlist[key])):

sql=f"select phone, lng, lat from customer where name='{orderlist[key][i]['name']}'"

cur.execute(sql)

rows=cur.fetchall()

lists[key][i]['phone']=rows[0]['phone']

lists[key][i]['lng']=rows[0]['lng']

lists[key][i]['lat']=rows[0]['lat']

return render\_template("orderprint.html", orderlist=lists, type='delivery', name=session['name'])

-로그인 한 사용자가 배달부일 경우 orderlist에 저장된 order와 사용자 정보를 주문 시간을 key로 하여 출력