Public.py

from flask import \*

from database import \*

public=Blueprint('public',\_\_name\_\_)

@public.route('/')

def main\_home():

return render\_template("main\_home.html")

@public.route('/login',methods=['get','post'])

def login():

if 'Login' in request.form:

username=request.form['username']

password=request.form['Password']

q="select \* from login where(username='%s' and password='%s')"%(username,password)

res=select(q)

if res:

session['login\_d']=res[0]['login\_id']

if res[0]['user\_type']=="admin":

return redirect(url\_for('admin.admin\_home'))

return render\_template('login.html')

admin.py

from flask import \*

from database import \*

admin=Blueprint('admin',\_\_name\_\_)

@admin.route('/admin\_home',methods=['get','post'])

def admin\_home():

return render\_template('admin\_home.html')

@admin.route('/admin\_manage\_vehicles',methods=['get','post'])

def admin\_manage\_vehicles():

data={}

q="SELECT \* FROM `vehicles` inner join fuel\_type using(type\_id)"

res=select(q)

data['vehicles']=res

q1="select \* from fuel\_type"

res1=select(q1)

data['fuel\_type']=res1

if 'action' in request.args:

action=request.args['action']

vid=request.args['vid']

# print(vid,"Value....\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

else:

action=None

if action=='update':

q="SELECT \* FROM `vehicles` inner join fuel\_type using(type\_id) WHERE `vehicle\_id`='%s'"%(vid)

res=select(q)

data['updatess']=res

if action=='remove':

q="DELETE FROM `vehicles` WHERE `vehicle\_id`='%s'"%(vid)

delete(q)

flash('removed')

return redirect(url\_for('admin.admin\_manage\_vehicles'))

if 'submitss' in request.form:

regnum=request.form['regnum']

vehicle=request.form['vehicle']

capacity=request.form['capacity']

q="UPDATE `vehicles` SET `regnum`='%s',`vehicle`='%s',`capacity`='%s' WHERE `vehicle\_id`='%s'"%(regnum,vehicle,capacity,vid)

update(q)

flash('successfully updated')

return redirect(url\_for('admin.admin\_manage\_vehicles'))

if 'submit' in request.form:

fuel=request.form['fuel']

regnum=request.form['regnum']

vehicle=request.form['vehicle']

capacity=request.form['capacity']

q="INSERT INTO `vehicles`(`type\_id`,`regnum`,`vehicle`,`capacity`)VALUES('%s','%s','%s','%s')"%(fuel,regnum,vehicle,capacity)

vid=insert(q)

q="INSERT INTO `stock` VALUES(NULL,'%s','0',NOW())"%(vid)

insert(q)

flash('success')

return redirect(url\_for('admin.admin\_manage\_vehicles'))

return render\_template('admin\_manage\_vehicles.html',data=data)

@admin.route('/admin\_manage\_fuel\_type',methods=['get','post'])

def admin\_manage\_fuel\_type():

data={}

q="SELECT \* FROM `fuel\_type`"

res=select(q)

data['fuel']=res

if 'action' in request.args:

action=request.args['action']

tid=request.args['tid']

else:

action=None

if action=='update':

q="SELECT \* FROM `fuel\_type` WHERE `type\_id`='%s'"%(tid)

res=select(q)

data['upftype']=res

if action=='remove':

q="DELETE FROM `fuel\_type` WHERE `type\_id`='%s'"%(tid)

delete(q)

flash('removed')

return redirect(url\_for('admin.admin\_manage\_fuel\_type'))

if 'utype' in request.form:

ftype=request.form['ftype']

rate=request.form['rate']

q="UPDATE `fuel\_type` SET `type\_name`='%s',`rate`='%s',`date\_time`=NOW() WHERE `type\_id`='%s'"%(ftype,rate,tid)

update(q)

flash('successfully updated')

return redirect(url\_for('admin.admin\_manage\_fuel\_type'))

if 'type' in request.form:

ftype=request.form['ftype']

rate=request.form['rate']

q="INSERT INTO `fuel\_type` VALUES(NULL,'%s','%s',NOW())"%(ftype,rate)

insert(q)

flash('success')

return redirect(url\_for('admin.admin\_manage\_fuel\_type'))

return render\_template('admin\_manage\_fuel\_type.html',data=data)

@admin.route('/admin\_view\_registered\_users',methods=['get','post'])

def admin\_view\_registered\_users():

data={}

q="SELECT \* FROM `users`"

res=select(q)

data['users']=res

return render\_template('admin\_view\_registered\_users.html',data=data)

@admin.route('/admin\_manage\_drivers',methods=['get','post'])

def admin\_manage\_drivers():

data={}

q="SELECT \*,CONCAT(`first\_name`,' ',`last\_name`)AS `d\_name` FROM `drivers`"

res=select(q)

data['drivers']=res

if 'action' in request.args:

action=request.args['action']

did=request.args['did']

else:

action=None

if action=='update':

q="SELECT \* FROM `drivers` WHERE `driver\_id`='%s'"%(did)

res=select(q)

data['updatess']=res

if action=='remove':

q="DELETE FROM `drivers` WHERE `driver\_id`='%s'"%(did)

delete(q)

flash('removed')

return redirect(url\_for('admin.admin\_manage\_drivers'))

if 'submitss' in request.form:

fname=request.form['fname']

lname=request.form['lname']

phone=request.form['phone']

email=request.form['email']

lnum=request.form['lnum']

q="UPDATE `drivers` SET `first\_name`='%s',`last\_name`='%s',`phone`='%s',`email`='%s',`licensenum`='%s' WHERE `driver\_id`='%s'"%(fname,lname,phone,email,lnum,did)

update(q)

flash('successfully updated')

return redirect(url\_for('admin.admin\_manage\_drivers'))

if 'submit' in request.form:

fname=request.form['fname']

lname=request.form['lname']

phone=request.form['phone']

email=request.form['email']

lnum=request.form['lnum']

uname=request.form['uname']

pwd=request.form['pwd']

q1="INSERT INTO `login`(`username`,`password`,`user\_type`)VALUES('%s','%s','driver')"%(uname,pwd)

res=insert(q1)

data['login']=res

q="INSERT INTO `drivers`(`login\_id`,`first\_name`,`last\_name`,`phone`,`email`,`licensenum`)VALUES('%s','%s','%s','%s','%s','%s')"%(res,fname,lname,phone,email,lnum)

insert(q)

flash('success')

return redirect(url\_for('admin.admin\_manage\_drivers'))

return render\_template('admin\_manage\_drivers.html',data=data)

@admin.route('/admin\_assign\_vehicles',methods=['get','post'])

def admin\_assign\_vehicles():

data={}

did=request.args['did']

name=request.args['name']

data['name']=name

q1="SELECT \* FROM `vehicles`"

res=select(q1)

data['vehicles']=res

q="SELECT \*,CONCAT(`first\_name`,' ',`last\_name`)AS `d\_name` FROM `drivers`WHERE `driver\_id`='%s'"%(did)

res=select(q)

data['drivers']=res

if 'submits' in request.form:

v\_id=request.form['vname']

dname=request.form['dname']

latitude=request.form['latitude']

longitude=request.form['longitude']

q="SELECT \* FROM `assign\_vehicle` WHERE `driver\_id`='%s'"%(did)

rs=select(q)

if rs:

flash("Already Alloted")

return redirect(url\_for('admin.admin\_manage\_drivers'))

else:

q="INSERT INTO `assign\_vehicle`(`vehicle\_id`,`driver\_id`,`latitude`,`longitude`)VALUES('%s','%s','%s','%s')"%(v\_id,did,latitude,longitude)

res=insert(q)

flash('success')

return redirect(url\_for('admin.admin\_manage\_drivers'))

return render\_template('admin\_assign\_vehicles.html',data=data)

@admin.route('/admin\_view\_assigned\_vehicle')

def admin\_view\_assigned\_vehicle():

data={}

did=request.args['did']

q="SELECT \* FROM `assign\_vehicle` INNER JOIN `vehicles` USING(`vehicle\_id`) INNER JOIN `drivers` USING(`driver\_id`)WHERE `driver\_id`='%s'"%(did)

res=select(q)

data['assign']=res

return render\_template('admin\_view\_assigned\_vehicle.html',data=data)

@admin.route('/admin\_view\_bookings',methods=['get','post'])

def admin\_view\_bookings():

data={}

q="SELECT \*,CONCAT(`drivers`.`first\_name`,' ',`drivers`.`last\_name`)AS `d\_name`,CONCAT(`users`.`first\_name`,' ',`users`.`last\_name`)AS `u\_name` FROM `bookings` INNER JOIN `users` USING(`user\_id`) INNER JOIN `drivers` USING(`driver\_id`)"

res=select(q)

data['bookings']=res

return render\_template('admin\_view\_bookings.html',data=data)

@admin.route('/admin\_view\_payments',methods=['get','post'])

def admin\_view\_payments():

data={}

q="SELECT \*,CONCAT(users.`first\_name`,' ',users.`last\_name`)AS `u\_name`,CONCAT(drivers.`first\_name`,' ',drivers.`last\_name`)AS `d\_name` FROM `payments` INNER JOIN `bookings` USING(`booking\_id`) INNER JOIN `users` USING(`user\_id`) INNER JOIN `drivers` USING(`driver\_id`)"

res=select(q)

data['payments']=res

return render\_template('admin\_view\_payments.html',data=data)

@admin.route('/admin\_view\_rating\_and\_review',methods=['get','post'])

def admin\_view\_rating\_and\_review():

data={}

q="SELECT \*,CONCAT(`users`.`first\_name`,' ',`users`.`last\_name`)AS `u\_name`,CONCAT(`drivers`.`first\_name`,' ',`drivers`.`last\_name`)AS `d\_name` FROM `rating` INNER JOIN `bookings` USING(`booking\_id`) INNER JOIN `users` USING(`user\_id`)INNER JOIN `drivers` USING(`driver\_id`)"

res=select(q)

data['rating']=res

return render\_template('admin\_view\_rating\_and\_review.html',data=data)

@admin.route('/admin\_view\_complaints',methods=['get','post'])

def admin\_view\_complaints():

data={}

q="SELECT \*,CONCAT(`first\_name`,' ',`last\_name`)AS `name` FROM `complaints` INNER JOIN `users` USING(`user\_id`)"

res=select(q)

data['complaints']=res

j=0

for i in range(1,len(res)+1):

print('submit'+str(i))

if 'submit'+str(i) in request.form:

reply=request.form['reply'+str(i)]

print(reply)

print(j)

print(res[j]['complaint\_id'])

q="update complaints set reply='%s' where complaint\_id='%s'" %(reply,res[j]['complaint\_id'])

print(q)

update(q)

flash("success")

return redirect(url\_for('admin.admin\_view\_complaints'))

j=j+1

return render\_template('admin\_view\_complaints.html',data=data)

android : User

from flask import \*

from database import \*

import demjson

import uuid

api=Blueprint('api',\_\_name\_\_)

@api.route('/login',methods=['get','post'])

def login():

data={}

username = request.args['username']

password = request.args['password']

q="SELECT \* from login where username='%s' and password='%s'" % (username,password)

res = select(q)

if res :

data['status'] = 'success'

data['data'] = res

data['method']='login'

else:

data['status'] = 'failed'

data['method']='login'

return demjson.encode(data)

@api.route('/userregister',methods=['get','post'])

def userregister():

data = {}

fname=request.args['fname']

lname=request.args['lname']

phone=request.args['phone']

licence=request.args['licence']

email=request.args['email']

uname=request.args['uname']

passw=request.args['pass']

lati=request.args['lati']

longi=request.args['longi']

q1="SELECT \* FROM login WHERE username='%s'" %(uname)

print(q1)

r=select(q1)

print(r)

if r:

data['status']='duplicate'

data['method']='userregister'

else:

q= "INSERT INTO `login` VALUES(NULL,'%s','%s','user')"%(uname,passw)

lid = insert(q)

qr="INSERT INTO `users` VALUES(NULL,'%s','%s','%s','%s','%s','%s','%s','%s')"%(lid,fname,lname,phone,email,licence,lati,longi)

id=insert(qr)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method']='userregister'

return demjson.encode(data)

@api.route('/updatepasslocation',methods=['get','post'])

def updatepasslocation():

data={}

latti=request.args['latti']

longi=request.args['longi']

logid=request.args['log\_id']

q="update `users` set `latitude`='%s',`longitude`='%s' where `login\_id`='%s'"%(latti,longi,logid)

id=update(q)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'updatepasslocation'

return demjson.encode(data)

@api.route('/user\_view\_nearest\_vehicle',methods=['get','post'])

def user\_view\_nearest\_vehicle():

data={}

data['method']='user\_view\_nearest\_vehicle'

lati=request.args['lati']

logi=request.args['logi']

# q = "SELECT \*,(3959 \* ACOS ( COS ( RADIANS('%s') ) \* COS( RADIANS( latitude) ) \* COS( RADIANS( longitude ) - RADIANS('%s') ) + SIN ( RADIANS('%s') ) \* SIN( RADIANS(latitude ) ))) AS user\_distance FROM colleges inner join ranking using(college\_id) HAVING user\_distance<31.068 ORDER BY rank desc" % (lati,logi,lati)

q="SELECT \*,(3959 \* ACOS ( COS ( RADIANS('%s') ) \* COS( RADIANS( latitude) ) \* COS( RADIANS( longitude ) - RADIANS('%s') ) + SIN ( RADIANS('%s') ) \* SIN( RADIANS(latitude ) ))) AS user\_distance,CONCAT(`drivers`.`first\_name`,' ',`drivers`.`last\_name`) AS dname FROM `vehicles` INNER JOIN `assign\_vehicle` USING(`vehicle\_id`) INNER JOIN `drivers` USING(`driver\_id`) INNER JOIN `fuel\_type` USING(`type\_id`) HAVING user\_distance<31.068 " % (lati,logi,lati)

print(q)

res=select(q)

if res:

data['status']='success'

data['data']=res

else:

data['status']='failed'

return demjson.encode(data)

@api.route('/user\_send\_fuel\_request',methods=['get','post'])

def user\_send\_fuel\_request():

data={}

type\_id=request.args['type\_id']

driver=request.args['driver']

login\_id=request.args['login\_id']

no\_titter=request.args['no\_titter']

tot\_rate=request.args['tot\_rate']

q="INSERT INTO `bookings` VALUES(NULL,(SELECT `user\_id` FROM `users` WHERE `login\_id`='%s'),'%s','%s','%s','%s',NOW(),'Pending')"%(login\_id,driver,type\_id,no\_titter,tot\_rate)

id=insert(q)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'user\_send\_fuel\_request'

return demjson.encode(data)

@api.route('/user\_view\_request',methods=['get','post'])

def user\_view\_request():

data={}

data['method']='user\_view\_request'

login\_id=request.args['login\_id']

q="SELECT \*,CONCAT(`drivers`.`first\_name`,' ',`drivers`.`last\_name`) AS driver FROM `bookings` INNER JOIN `drivers` ON `drivers`.`driver\_id`=`bookings`.`driver\_id` INNER JOIN `fuel\_type` ON `fuel\_type`.`type\_id`=`bookings`.`type\_id` INNER JOIN `vehicles` ON `vehicles`.`type\_id`=`fuel\_type`.`type\_id` WHERE `bookings`.`user\_id`=(SELECT `user\_id` FROM `users` WHERE `login\_id`='%s')"%(login\_id)

res=select(q)

if res:

data['status']='success'

data['data']=res

else:

data['status']='failed'

return demjson.encode(data)

@api.route('/user\_payment',methods=['get','post'])

def user\_payment():

data={}

booking\_id=request.args['booking\_id']

q1="UPDATE `bookings` SET `status`='Paid' WHERE `booking\_id`='%s'"%(booking\_id)

update(q1)

q="INSERT INTO `payments` VALUES(NULL,'%s',NOW())"%(booking\_id)

id=insert(q)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'user\_payment'

return demjson.encode(data)

@api.route('/user\_rate\_fuel',methods=['get','post'])

def user\_rate\_fuel():

data={}

booking\_id=request.args['booking\_id']

ratings=request.args['rating']

review=request.args['review']

q="SELECT \* FROM `rating` WHERE `booking\_id`='%s'"%(booking\_id)

res=select(q)

if res:

q="UPDATE `rating` SET `rating`='%s',`review`='%s',`date\_time`=CURDATE() WHERE `booking\_id`='%s'"%(ratings,review,booking\_id)

update(q)

data['method'] = 'user\_rate\_fuel'

else:

q="INSERT INTO `rating` VALUES(NULL,'%s','%s','%s',CURDATE())"%(booking\_id,ratings,review)

id=insert(q)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'user\_rate\_fuel'

return demjson.encode(data)

@api.route('/user\_view\_rated',methods=['get','post'])

def user\_view\_rated():

data = {}

booking\_id=request.args['booking\_id']

q=" SELECT \* FROM `rating` WHERE `booking\_id`='%s'"%(booking\_id)

print(q)

result=select(q)

if result:

data['status'] = 'success'

data['data'] = result[0]['rating']

data['data1'] = result[0]['review']

else:

data['status'] = 'failed'

data['method'] = 'user\_view\_rated'

return demjson.encode(data)

@api.route('/user\_send\_complaints',methods=['get','post'])

def user\_send\_complaints():

data = {}

loginid=request.args['loginid']

complaints=request.args['complaints']

qr="INSERT INTO `complaints` VALUES(NULL,(SELECT `user\_id` FROM `users` WHERE `login\_id`='%s'),'%s',NOW(),'pending')"%(loginid,complaints)

print(qr)

id=insert(qr)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method']='user\_send\_complaints'

return demjson.encode(data)

@api.route('/user\_view\_complaints',methods=['get','post'])

def user\_view\_complaints():

data={}

loginid=request.args['loginid']

q="SELECT \* FROM `complaints` WHERE `user\_id`=(SELECT `user\_id` FROM `users` WHERE `login\_id`='%s')"%(loginid)

res = select(q)

if res :

data['status'] = 'success'

data['data'] = res

else:

data['status'] = 'failed'

data['method']='user\_view\_complaints'

return demjson.encode(data)

android : Driver

@api.route('/driver\_view\_assigned\_vehicle',methods=['get','post'])

def driver\_view\_assigned\_vehicle():

data = {}

login\_id=request.args['login\_id']

q="SELECT \* FROM `assign\_vehicle` INNER JOIN `vehicles` USING(`vehicle\_id`) INNER JOIN `stock` USING(`vehicle\_id`) WHERE `driver\_id`=(SELECT `driver\_id` FROM `drivers` WHERE `login\_id`='%s')"%(login\_id)

result=select(q)

if result:

data['status'] = 'success'

data['data'] = result

else:

data['status'] = 'failed'

data['method'] = 'driver\_view\_assigned\_vehicle'

return demjson.encode(data)

@api.route('/driver\_view\_stock',methods=['get','post'])

def driver\_view\_stock():

data = {}

vehicle\_id=request.args['vehicle\_id']

q="SELECT \* FROM `stock` WHERE `vehicle\_id`='%s'"%(vehicle\_id)

# q="SELECT \* FROM `drivers` INNER JOIN `assign\_vehicle` USING(`driver\_id`) INNER JOIN `stock` USING(`vehicle\_id`) WHERE `driver\_id`=(SELECT `driver\_id` FROM `drivers` WHERE `login\_id`='%s')"%(login\_id)

result=select(q)

if result:

data['status'] = 'success'

data['stock\_id'] = result[0]['stock\_id']

data['stock'] = result[0]['stock']

else:

data['status'] = 'failed'

data['method'] = 'driver\_view\_stock'

return demjson.encode(data)

@api.route('/driver\_update\_stock',methods=['get','post'])

def driver\_update\_stock():

data={}

stock=request.args['stock']

vehicle\_id=request.args['vehicle\_id']

q1="UPDATE `stock` SET `stock`='%s' WHERE `vehicle\_id`='%s'"%(stock,vehicle\_id)

id=update(q1)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'driver\_update\_stock'

return demjson.encode(data)

@api.route('/driver\_view\_request',methods=['get','post'])

def driver\_view\_request():

data = {}

login\_id=request.args['login\_id']

q="SELECT \*,CONCAT(`users`.`first\_name`,' ',`users`.`last\_name`) AS username FROM `bookings` INNER JOIN `users` USING(`user\_id`) INNER JOIN `fuel\_type` USING(`type\_id`) INNER JOIN `assign\_vehicle` USING(`driver\_id`) WHERE `driver\_id`=(SELECT `driver\_id` FROM `drivers` WHERE `login\_id`='%s')"%(login\_id)

result=select(q)

if result:

data['status'] = 'success'

data['data'] = result

else:

data['status'] = 'failed'

data['method'] = 'driver\_view\_request'

return demjson.encode(data)

@api.route('/driver\_accept\_request',methods=['get','post'])

def driver\_accept\_request():

data={}

booking\_ids=request.args['booking\_ids']

q1=" UPDATE `bookings` SET `status`='Accept' WHERE `booking\_id`='%s'"%(booking\_ids)

id=update(q1)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'driver\_accept\_request'

return demjson.encode(data)

@api.route('/driver\_accept\_payment',methods=['get','post'])

def driver\_accept\_payment():

data={}

booking\_ids=request.args['booking\_ids']

vehicle\_ids=request.args['vehicle\_ids']

nooflitters=request.args['nooflitters']

q2="UPDATE `stock` SET `stock`=`stock`-'%s' WHERE `vehicle\_id`='%s'"%(nooflitters,vehicle\_ids)

update(q2)

q1="UPDATE `bookings` SET `status`='Payment Received' WHERE `booking\_id`='%s'"%(booking\_ids)

id=update(q1)

if id>0:

data['status'] = 'success'

else:

data['status'] = 'failed'

data['method'] = 'driver\_accept\_payment'

return demjson.encode(data)

@api.route('/driver\_view\_ratings',methods=['get','post'])

def driver\_view\_ratings():

data = {}

booking\_id=request.args['booking\_id']

q=" SELECT \* FROM `rating` WHERE `booking\_id`='%s'"%(booking\_id)

print(q)

result=select(q)

if result:

data['status'] = 'success'

data['data'] = result[0]['rating']

data['data1'] = result[0]['review']

else:

data['status'] = 'failed'

data['method'] = 'driver\_view\_ratings'

return demjson.encode(data)