**SPRINT – 4 Code**

from flask import Blueprint,render\_template,redirect,url\_for,session,request,flash

from database import \*

from core import \*

import demjson

import random

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

@voter.route('/home/',methods=['get','post'])

def home():

data = {}

return render\_template('voter\_home.html',data=data)

@voter.route('/view\_booths/',methods=['get','post'])

def view\_booths():

data = {}

login\_id=session['login\_id']

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

q="select \* from election"

election=select(q)

data['election']=election

if 'submit' in request.form:

election\_id=request.form['election\_id']

q="""SELECT \* FROM voter INNER JOIN booth USING(booth\_id)

INNER JOIN constituency USING(constituency\_id)

INNER JOIN state USING(state\_id)

INNER JOIN election\_details USING(state\_id) INNER JOIN election USING(election\_id)

WHERE voter\_id=(select voter\_id from voter where login\_id='%s') AND election\_id='%s'"""%(login\_id,election\_id)

print(q)

data['election\_details']=select(q)

return render\_template('voter\_view\_booths.html',data=data)

@voter.route('/view\_candidates/',methods=['get','post'])

def view\_candidates():

data = {}

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

action=None

q="SELECT \* FROM candidate INNER JOIN constituency USING (constituency\_id) WHERE candidate\_status='approved'"

candidate=select(q)

data['candidate']=candidate

q="SELECT \* FROM promise INNER JOIN candidate USING(candidate\_id)"

promises=select(q)

data['promises']=promises

return render\_template('voter\_view\_candidates.html',data=data)

@voter.route('/view\_results',methods=['get','post'])

def view\_results():

data = {}

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

action=None

q="SELECT \* FROM INNER JOIN promises USING(promise\_id)"

promises=select(q)

data['promises']=promises

return render\_template('voter\_view\_candidates.html',data=data)

@voter.route('/make\_complaint/',methods=['get','post'])

def make\_complaint():

data={}

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

action=None

login\_id=session['login\_id']

q="select \* from voter where login\_id='%s'"%(login\_id)

print(q)

voter = select(q)

data['voter']=voter

voter\_id=data['voter'][0]['voter\_id']

print(q)

res=select(q)

print(res)

# voter\_id=res[0]['voter\_id']

if 'add\_complaint' in request.form:

# voter\_id=request.form['voter\_id']

description=request.form['description']

# reply=request.form['reply']

# complaint\_date=request.form['complaint\_date']

q = "insert into complaint(user\_id,description,reply,complaint\_date,user\_type)VALUES('%s','%s','pending',curdate(),'voter')" % (voter\_id,description)

insert(q)

flash('Success')

if 'update\_complaint' in request.form:

complaint\_id=request.form['complaint\_id']

description=request.form['description']

#reply=request.form['reply']

#complaint\_date=request.form['complaint\_date']

q="update complaint set description='%s' WHERE complaint\_id='%s' " % (description,complaint\_id)

update(q)

flash('Success')

if action=='update':

id = request.args['id']

q = "select \* from complaint where complaint\_id='%s'" % (id)

update\_data = select(q)

data['update\_data']=update\_data

if action=='remove':

q = "delete from complaint where complaint\_id='%s'" % (id)

delete(q)

flash('Deleted')

q="SELECT \* FROM complaint INNER JOIN voter ON voter\_id=user\_id AND user\_type='voter' WHERE voter\_id='%s'"%(voter\_id)

complaint = select(q)

data['complaint']=complaint

return render\_template('voter\_make\_complaint.html',data=data)

@voter.route('/cast\_vote',methods=['get','post'])

def cast\_vote():

data = {}

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

action=None

login\_id=session['login\_id']

print(login\_id)

q="select \* from voter where login\_id='%s'"%(login\_id)

voter = select(q)

data['voter']=voter

voter\_id=voter[0]['voter\_id']

q="""

SELECT \* FROM election

INNER JOIN election\_details USING(election\_id)

INNER JOIN constituency USING(state\_id)

INNER JOIN booth USING(constituency\_id)

INNER JOIN voter USING(booth\_id)

WHERE polling\_date=CURDATE() AND voter\_id='%s' GROUP BY election\_id

"""%(voter\_id)

print(q)

election=select(q)

data['election']=election

q="SELECT \* FROM state"

state=select(q)

data['state']=state

q="select \* from constituency"

constituency=select(q)

data['constituency']=constituency

if 'continue' in request.form:

election\_id=request.form['election\_id']

q="""SELECT \* FROM election

INNER JOIN election\_details USING(election\_id)

INNER JOIN constituency USING(state\_id)

INNER JOIN booth USING(constituency\_id)

INNER JOIN voter USING(booth\_id)

INNER JOIN candidate USING(constituency\_id,election\_id)

INNER JOIN party USING(party\_id)

LEFT JOIN(SELECT election\_id,voting\_status FROM polling)temp USING(election\_id)

WHERE polling\_date=CURDATE() AND voter\_id='%s' AND election\_id='%s'"""%(voter\_id,election\_id)

candidate=select(q)

print(q)

data['candidates']=candidate

return render\_template('voter\_cast\_vote.html',data=data)

@voter.route('/make\_vote/',methods=['get','post'])

def make\_vote():

data={}

candidate\_id=request.args['Candidate\_id']

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

login\_id=session['login\_id']

q="select \* from voter where login\_id='%s'"%login\_id

voter=select(q)

voter\_id=voter[0]['voter\_id']

email=voter[0]['email']

otp=random.randint(1000,9999)

if 'continue' in request.form:

otp=request.form['otp']

q="select \* from polling where otp='%s' and voter\_id='%s' and voting\_status='pending'"%(otp,voter\_id)

res=select(q)

print(q)

if res:

q="update polling set voting\_status='voted' where polling\_id='%s'"%(res[0]['polling\_id'])

update(q)

flash('voting successfull')

return redirect(url\_for('voter.home'))

else:

flash('incorrect OTP,please try again')

else:

q="INSERT INTO `polling`(`voter\_id`, `candidate\_id`, `datetime`, `otp`, `voting\_status`,election\_id) VALUES('%s','%s',curdate(),'%s','pending','%s')"%(voter\_id,candidate\_id,str(otp),election\_id)

insert(q)

sent\_mail(email,'otp for voting','your OTP :'+str(otp))

flash('please check your mail for OTP')

return render\_template('voter\_make\_vote.html',data=data)

@voter.route('/view\_result/',methods=['get','post'])

def view\_result():

data={}

if 'action' in request.args:

action = request.args['action']

id = request.args['id']

else:

action=None

q="SELECT \* FROM election INNER JOIN candidate\_count USING(election\_id) GROUP BY election\_id"

data['election']=select(q)

if 'submit' in request.form:

election\_id=request.form['election\_id']

login\_id=session['login\_id']

q="select \* from voter where login\_id='%s'"%login\_id

voter=select(q)

booth\_id=voter[0]['booth\_id']

q="""SELECT \* FROM (SELECT candidate\_id,MAX(vote\_count) AS vote\_count,election\_id,constituency\_id FROM

(SELECT candidate\_id,CURDATE(),COUNT(candidate\_id) AS vote\_count,election\_id,constituency\_id FROM polling

INNER JOIN (SELECT candidate\_id,constituency\_id FROM candidate)tbl USING(candidate\_id)

WHERE election\_id=%s GROUP BY candidate\_id)tbl1 GROUP BY constituency\_id)tbl2

INNER JOIN candidate USING(candidate\_id,constituency\_id) INNER JOIN party USING(party\_id)

INNER JOIN constituency USING(constituency\_id)

INNER JOIN state USING(state\_id)

inner join booth using(constituency\_id) where booth\_id='%s'"""%(election\_id,booth\_id)

candidate\_count = select(q)

data['candidate\_count']=candidate\_count

return render\_template('voter\_view\_result.html',data=data)

@voter.route('/logout/',methods=['get','post'])

def logout():

data={}

session.clear()

return redirect(url\_for('public.login'))

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

def function():

data={}

action=request.args['action']

if action=='get\_states':

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

q="""SELECT \* FROM election\_details

INNER JOIN election USING(election\_id)

INNER JOIN state USING(state\_id) WHERE election\_id='%s'"""%(election\_id)

states=select(q)

return demjson.encode(states)

if action=='get\_constituency':

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

q="""SELECT \* FROM constituency

INNER JOIN state USING(state\_id) WHERE state\_id='%s'"""%(state\_id)

constituency=select(q)

return demjson.encode(constituency)