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

from hired import app, db, bcrypt

from hired.forms import RegistrationForm, LoginForm, UpdateAccountForm, RecUpdateAccountForm, CreateJobForm, \

SelectJobForm

from hired.models import User, Recruiter, Job, Job\_User

from flask\_login import login\_user, current\_user, logout\_user, login\_required

import urllib

# from hired.import\_csv import recruiter\_list

posts = [

{

'title': 'Join Us @ Amazon',

'author': 'Amazon',

'date\_posted': '22/01/20',

'content': 'Looking for fresh talent. \n Join our family. \n Everyone has a place here. \n We make the world a better place.'

},

{

'title': 'Looking to start fresh',

'author': 'Enakshi Jamwal',

'date\_posted': '22/01/20',

'content': 'Recently graduated from MPSTME. \n Specialise in UX Design. \n Excited to start working. \n Ready to join as soon as quarantine is over.'

}

]

@app.route('/')

@app.route('/home')

def home():

return render\_template('home.html', posts=posts)

@app.route('/about')

def about():

return render\_template('about.html', title='About Us')

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

def register():

if current\_user.is\_authenticated:

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

form = RegistrationForm()

if form.validate\_on\_submit():

hashed\_password = bcrypt.generate\_password\_hash(form.password.data).decode('utf-8')

user = User(username=form.username.data, email=form.email.data, password=hashed\_password)

db.session.add(user)

db.session.commit()

login\_user(user, remember=form.remember.data)

flash('Your account has been created. You can now enter your information.', 'success')

return redirect(url\_for('account'))

return render\_template('register.html', title='Register', form=form)

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

def login():

if current\_user.is\_authenticated:

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

form = LoginForm()

if form.validate\_on\_submit():

user = User.query.filter\_by(email=form.email.data).first()

if user and bcrypt.check\_password\_hash(user.password, form.password.data):

login\_user(user, remember=form.remember.data)

next\_page = request.args.get('next')

# subdomain = request.url.hostname.split('.')[0]

return redirect(next\_page) if next\_page else redirect(url\_for('home'))

else:

flash('Login Unsuccessful. Please check email and password.', 'danger')

return render\_template('login.html', title='Login', form=form)

@app.route('/logout')

def logout():

logout\_user()

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

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

@login\_required

def account():

form = UpdateAccountForm()

if form.validate\_on\_submit():

current\_user.username = form.username.data

current\_user.email = form.email.data

db.session.commit()

flash('Your account has been updated!', 'success')

return redirect(url\_for('account'))

elif request.method == 'GET':

form.username.data = current\_user.username

form.email.data = current\_user.email

form.degree\_type.data = current\_user.degree\_type

form.experience.data = current\_user.experience

form.major.data = current\_user.major

form.skill\_1.data = current\_user.skill\_1

form.skill\_1\_level.data = current\_user.skill\_1\_level

form.dbms.data = current\_user.dbms

# image\_file = url\_for('static', filename='profile\_pics/' + current\_user.image\_file)

image\_file = url\_for('static', filename='default.jpg')

return render\_template('account.html', title='Account', image\_file=image\_file, form=form)

@app.route('/user\_list')

def user\_list():

users = User.query.all()

return render\_template('user\_list.html', title='Candidates List', users=users)

@app.route('/job\_list')

def job\_list():

jobs = Job.query.all()

return render\_template('job\_list.html', title='Jobs List', jobs=jobs)

@app.route('/recruiter\_list')

def recruiter\_list():

recruiters = Recruiter.query.all()

return render\_template('recruiter\_list.html', title='Recruiters List', recruiters=recruiters)

@app.route('/', subdomain='recruit')

@app.route('/home', subdomain='recruit')

def home\_rec():

return render\_template('home\_rec.html')

@app.route('/test', subdomain='recruit')

def test():

return render\_template('test.html', title='Test')

@app.route('/recruiter\_list', subdomain='recruit')

def recruiter\_list\_rec():

recruiters = Recruiter.query.all()

return render\_template('recruiter\_list.html', title='Recruiters List', recruiters=recruiters)

@app.route('/register', subdomain='recruit', methods=['GET', 'POST'])

def register\_rec():

if current\_user.is\_authenticated:

return redirect(url\_for('home\_rec'))

form = RegistrationForm()

if form.validate\_on\_submit():

hashed\_password = bcrypt.generate\_password\_hash(form.password.data).decode('utf-8')

recruiter = Recruiter(username=form.username.data, email=form.email.data, password=hashed\_password)

db.session.add(recruiter)

db.session.commit()

login\_user(recruiter, remember=form.remember.data)

flash('Your account has been created. You can now enter your information.', 'success')

return redirect(url\_for('account\_rec'))

return render\_template('register\_rec.html', title='Register', form=form)

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

def login\_rec():

if current\_user.is\_authenticated:

return redirect(url\_for('home\_rec'))

form = LoginForm()

if form.validate\_on\_submit():

recruiter = Recruiter.query.filter\_by(email=form.email.data).first()

if recruiter and bcrypt.check\_password\_hash(recruiter.password, form.password.data):

login\_user(recruiter, remember=form.remember.data)

next\_page = request.args.get('next')

return redirect(next\_page) if next\_page else redirect(url\_for('home\_rec'))

else:

flash('Login Unsuccessful. Please check email and password.', 'danger')

return render\_template('login\_rec.html', title='Login', form=form)

@app.route('/logout', subdomain='recruit')

def logout\_rec():

logout\_user()

return redirect(url\_for('home\_rec'))

@app.route('/account', methods=['GET', 'POST'], subdomain='recruit')

@login\_required

def account\_rec():

form = RecUpdateAccountForm()

if form.validate\_on\_submit():

current\_user.username = form.username.data

current\_user.email = form.email.data

db.session.commit()

flash('Your account has been updated!', 'success')

return redirect(url\_for('account\_rec'))

elif request.method == 'GET':

form.username.data = current\_user.username

form.email.data = current\_user.email

form.location.data = current\_user.location

# image\_file = url\_for('static', filename='profile\_pics/' + current\_user.image\_file)

image\_file = url\_for('static', filename='default.jpg')

return render\_template('account\_rec.html', title='Account', image\_file=image\_file, form=form)

@app.route('/view\_jobs', methods=['GET'], subdomain='recruit')

@login\_required

def view\_jobs():

jobs = Job.query.filter\_by(company=current\_user).all()

return render\_template('job\_list.html', title='Jobs List', jobs=jobs)

@app.route('/create\_jobs', methods=['GET', 'POST'], subdomain='recruit')

@login\_required

def create\_jobs():

form = CreateJobForm()

if form.validate\_on\_submit():

job = Job(job\_id=form.job\_id.data, company=current\_user, skill=form.skill.data, location=form.location.data)

db.session.add(job)

db.session.commit()

flash('Your job has been created!', 'success')

return redirect(url\_for('account\_rec'))

image\_file = url\_for('static', filename='default.jpg')

return render\_template('create\_jobs.html', title='Create Jobs', image\_file=image\_file, form=form)

@app.route('/select\_jobs', methods=['GET', 'POST'])

@login\_required

def select\_jobs():

form = SelectJobForm()

print(form.job\_id.data)

jobs = Job.query.all()

if form.validate\_on\_submit():

job = Job.query.filter\_by(job\_id=form.job\_id.data).first()

job\_user = Job\_User()

job\_user.job\_id = job.job\_id

job\_user.user\_id = current\_user.get\_id()

# job\_user = Job\_User(job=job, user=current\_user)

db.session.add(job\_user)

db.session.commit()

flash('Your job has been selected!', 'success')

return redirect(url\_for('view\_jobs'))

return render\_template('select\_jobs.html', title='Select Jobs', form=form, jobs=jobs)

@app.route('/view\_jobs', methods=['GET', 'POST'])

@login\_required

def view\_jobs\_can():

job\_users = Job\_User.query.filter\_by(user\_id=current\_user.get\_id()).all()

lis = []

jobs = []

for ju in job\_users:

lis.append(ju.job\_id)

for x in lis:

jobs.append(Job.query.filter\_by(job\_id=x).first())

return render\_template('job\_list.html', title='Jobs List', jobs=jobs)

@app.route('/view\_candidates', methods=['GET', 'POST'], subdomain='recruit')

@login\_required

def view\_cans():

jobs = Job.query.filter\_by(company=current\_user).all()

lis = []

jus = []

for j in jobs:

lis.append(j.job\_id)

for x in lis:

jus.append(Job\_User.query.filter\_by(job\_id=x).first())

lis = []

cans = []

for ju in jus:

lis.append(ju.user\_id)

for x in lis:

cans.append(User.query.filter\_by(id=x).first())

return render\_template('view\_cans.html', title='Candidates List', users=cans)