if var\_t < 20:  
 return HttpResponse("Please update your skills")  
 else:  
 if request.session['php'] < 4:  
 return HttpResponse("Update your PHP skill")  
 if request.session['html'] < 4:  
 return HttpResponse("Update your HTML skill")  
 if request.session['css'] < 4:  
 return HttpResponse("Update your CSS skill")  
 if request.session['wpress'] < 4:  
 return HttpResponse("Update your WordPress skill")  
 elif request.session['bstrap'] < 4:  
 return HttpResponse("Update your Bootstrap skill")  
 else:  
 pass  
return HttpResponse("You can be a WebDeveloper")

if request.session['php'] < 4:  
 messages.success(request, "Improve your PHP skill")  
 return render(request, 'web/result.html')  
if request.session['html'] < 4:  
 messages.success(request, "Improve your HTML skill")  
 return render(request, 'web/result.html')  
if request.session['css'] < 4:  
 messages.success(request, "Improve your CSS skill")  
 return render(request, 'web/result.html')  
if request.session['wpress'] < 4:  
 messages.success(request, "Improve your Wordpress skill")  
 return render(request, 'web/result.html')  
elif request.session['bstrap'] < 4:  
 messages.success(request, "Improve your Bootstrap skill")  
 return render(request, 'web/result.html')  
else:  
 pass

class Question(models.Model):  
 q\_id = models.AutoField  
 category = models.CharField(max\_length=50)  
 question = models.TextField(max\_length=300)  
 option1 = models.CharField(max\_length=90)  
 option2 = models.CharField(max\_length=90)  
 option3 = models.CharField(max\_length=90)  
 option4 = models.CharField(max\_length=90)  
 answer = models.CharField(max\_length=90)  
  
 class Meta:  
 db\_table = 'question\_set'  
  
 def \_\_str\_\_(self):  
 return self.category

from django.shortcuts import render  
from django.contrib import messages  
from .models import \*  
  
  
# Create your views here.  
def home(request):  
 return render(request, 'application/home.html')  
  
  
def rating(request):  
 return render(request, 'application/rating.html')  
  
  
def calc(request):  
 global var\_fe, var\_ds  
 if request.method == 'GET':  
 request.session['php'] = int(request.GET.get('p'))  
 request.session['html'] = int(request.GET.get('h'))  
 request.session['css'] = int(request.GET.get('c'))  
 request.session['wpress'] = int(request.GET.get('w'))  
 request.session['bstrap'] = int(request.GET.get('b'))  
 request.session['r'] = int(request.GET.get('r'))  
 request.session['py'] = int(request.GET.get('py'))  
 request.session['stat'] = int(request.GET.get('stat'))  
 request.session['mcl'] = int(request.GET.get('ml'))  
 var\_fe = request.session['php'] + request.session['html'] + request.session['css'] + request.session['wpress'] + \  
 request.session['bstrap']  
 var\_ds = request.session['r'] + request.session['py'] + request.session['stat'] + request.session['mcl']  
 if var\_fe < 20 and var\_ds < 16:  
 messages.success(request, "You need to improve your skills")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if var\_fe < 20 and var\_ds >= 16 and var\_fe < var\_ds:  
 if request.session['r'] >= 4 and request.session['py'] >= 5 and request.session['stat'] >= 4 and \  
 request.session['mcl'] >= 4:  
 messages.success(request, "You can be a Data Scientist")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 else:  
 messages.success(request, "You need to improve your Data Science skills")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if var\_ds < 16 and var\_fe >= 20 and var\_ds < var\_fe:  
 if request.session['html'] >= 5 and request.session['css'] >= 5 and request.session['wpress'] >= 4 and \  
 request.session['bstrap'] >= 5 and request.session['php'] >= 4:  
 messages.success(request, "You can be a FrontEnd Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if request.session['php'] >= 5 and request.session['html'] >= 4 and request.session['css'] >= 4 and \  
 request.session['wpress'] >= 5 and request.session['bstrap'] >= 4:  
 messages.success(request, "You can be a BackEnd Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if request.session['php'] >= 5 and request.session['html'] >= 5 and request.session['css'] >= 5 and \  
 request.session['wpress'] >= 5 and request.session['bstrap'] >= 5:  
 messages.success(request, "You can be a Good Web Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 else:  
 messages.success(request, "You need to improve your Web Development skills")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if 16 <= var\_ds < var\_fe and var\_fe >= 20:  
 if request.session['html'] >= 5 and request.session['css'] >= 5 and request.session['wpress'] >= 4 and \  
 request.session['bstrap'] >= 5 and request.session['php'] >= 4:  
 messages.success(request, "You can be a FrontEnd Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if request.session['php'] >= 5 and request.session['html'] >= 4 and request.session['css'] >= 4 and \  
 request.session['wpress'] >= 5 and request.session['bstrap'] >= 4:  
 messages.success(request, "You can be a BackEnd Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if request.session['php'] >= 5 and request.session['html'] >= 5 and request.session['css'] >= 5 and \  
 request.session['wpress'] >= 5 and request.session['bstrap'] >= 5:  
 messages.success(request, "You can be a Good Web Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 else:  
 messages.success(request, "You need to improve your Web Development skills")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if 20 <= var\_fe < var\_ds and var\_ds >= 16:  
 messages.success(request, "You can be a Data Scientist")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
 if var\_ds == var\_fe:  
 messages.success(request, "You can be a Data Scientist or Web Developer")  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
  
 else:  
 pass  
 return render(request, 'application/result.html', {'score\_fe': var\_fe, 'score\_ds': var\_ds})  
  
  
def test(request):  
 question\_php = Php.objects.all()  
 return render(request, 'application/test.html', {'question\_php': question\_php})  
  
  
def test2(request):  
 question\_html = Html.objects.all()  
 return render(request, 'application/test2.html', {'question\_html': question\_html})  
  
  
def test3(request):  
 question\_css = Css.objects.all()  
 return render(request, 'application/test3.html', {'question\_css': question\_css})  
  
  
def test4(request):  
 question\_wp = Wordpress.objects.all()  
 return render(request, 'application/test4.html', {'question\_wp': question\_wp})  
  
  
def test5(request):  
 question\_bs = Bootstrap.objects.all()  
 return render(request, 'application/test5.html', {'question\_bs': question\_bs})  
  
  
def test6(request):  
 question\_r = Rtool.objects.all()  
 return render(request, 'application/test6.html', {'question\_r': question\_r})  
  
  
def test7(request):  
 question\_py = Python.objects.all()  
 return render(request, 'application/test7.html', {'question\_py': question\_py})  
  
  
def test8(request):  
 question\_st = Stat.objects.all()  
 return render(request, 'application/test8.html', {'question\_st': question\_st})  
  
  
def test9(request):  
 question\_ml = Machine.objects.all()  
 return render(request, 'application/test9.html', {'question\_ml': question\_ml})  
  
  
def result(request):  
 question\_php = Php.objects.all()  
 question\_html = Html.objects.all()  
 question\_css = Css.objects.all()  
 question\_wp = Wordpress.objects.all()  
 question\_bs = Bootstrap.objects.all()  
 question\_r = Rtool.objects.all()  
 question\_py = Python.objects.all()  
 question\_st = Stat.objects.all()  
 question\_ml = Machine.objects.all()  
 total\_php = score\_php(request, question\_php)  
 total\_html = score\_html(request, question\_html)  
 total\_css = score\_css(request, question\_css)  
 total\_wp = score\_wp(request, question\_wp)  
 total\_bs = score\_bs(request, question\_bs)  
 total\_r = score\_r(request, question\_r)  
 total\_py = score\_py(request, question\_py)  
 total\_st = score\_st(request, question\_st)  
 total\_ml = score\_ml(request, question\_ml)  
 total\_fe = total\_php + total\_html + total\_css + total\_wp + total\_bs  
 total\_ds = total\_r + total\_py + total\_ml + total\_st  
 if total\_ds <= 8 and total\_fe <= 10:  
 messages.success(request, "You need to improve your skills")  
 return render(request, 'application/result.html', {'score\_fe': total\_fe, 'score\_ds': total\_ds})  
 if total\_fe <= 10 and total\_ds > 8:  
 messages.success(request, "You can be a Data Scientist")  
 return render(request, 'application/result.html', {'score\_fe': total\_fe, 'score\_ds': total\_ds})  
 if total\_fe > 10 and total\_ds <= 8:  
 messages.success(request, "You can be a Web Developer")  
 return render(request, 'application/result.html', {'score\_fe': total\_fe, 'score\_ds': total\_ds})  
 if total\_fe > total\_ds > 8:  
 messages.success(request, "You can be a Web Developer")  
 return render(request, 'application/result.html', {'score\_fe': total\_fe, 'score\_ds': total\_ds})  
 if total\_ds > total\_fe > 10:  
 messages.success(request, "You can be a Data Scientist")  
 return render(request, 'application/result.html', {'score\_fe': total\_fe, 'score\_ds': total\_ds})  
  
  
def score\_php(request, question\_php):  
 total\_php = 0  
 for i in question\_php:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_php += 1  
 return total\_php  
  
  
def score\_html(request, question\_html):  
 total\_html = 0  
 for i in question\_html:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_html += 1  
 return total\_html  
  
  
def score\_css(request, question\_css):  
 total\_css = 0  
 for i in question\_css:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_css += 1  
 return total\_css  
  
  
def score\_wp(request, question\_wp):  
 total\_wp = 0  
 for i in question\_wp:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_wp += 1  
 return total\_wp  
  
  
def score\_bs(request, question\_bs):  
 total\_bs = 0  
 for i in question\_bs:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_bs += 1  
 return total\_bs  
  
  
def score\_r(request, question\_r):  
 total\_r = 0  
 for i in question\_r:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_r += 1  
 return total\_r  
  
  
def score\_py(request, question\_py):  
 total\_py = 0  
 for i in question\_py:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_py += 1  
 return total\_py  
  
  
def score\_st(request, question\_st):  
 total\_st = 0  
 for i in question\_st:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_st += 1  
 return total\_st  
  
  
def score\_ml(request, question\_ml):  
 total\_ml = 0  
 for i in question\_ml:  
 answer = i.answer  
 user\_answer = request.POST.get(str(i.answer))  
 if user\_answer == answer:  
 total\_ml += 1  
 return total\_ml