

Urls.py



CI Python Linter

```
1 from django.urls import path, include
2 from . import views
3
4 urlpatterns = [
5     path('', views.index),
6     path('create/user', views.createUser, name='createUser'),
7     path('user/info', views.getUser, name='getUser'),
8     path('employer/jobposts', views.retrieveEmployerJobPosts, name='jobposts'),
9     path('employer/jobpost/create', views.createJobPost),
10    path('employer/jobpost/update/<str:id>',
11         views.updateJobPost, name='updateJobPost'),
12    path('employer/jobpost/delete/<str:id>',
13         views.deleteJobPost, name='deleteJobPost'),
14    path('employer/jobpost/get/<str:id>',
15         views.getJobPostById, name='getJobPostById'),
16    path('employer/jobpost/<str:job_id>/application/<str:application_id>',
17         views.getApplication, name='getApplication'),
18    path('jobseeker/availableJobPosts',
19         views.retrieveAvailableJobPosts, name='availableJobPosts'),
20    path('jobseeker/applications', views.retrieveApplications,
21         name='applications'),
22    path('jobseeker/info/update', views.updateJobSeekerInfo,
23         name='updateJobSeekerInfo'),
24    path('jobseeker/retrieve/profile', views.getJobSeekerCv,
25         name='getJobSeekerCv'),
26    path('jobseeker/workexperience/create',
27         views.createWorkExperience, name='createWorkExperience'),
28    path('jobseeker/workexperience/delete/<str:id>',
29         views.deleteWorkExperience, name='deleteWorkExperience'),
30    path('jobseeker/workexperience/update/<str:id>',
31         views.updateWorkExperience, name='updateWorkExperience'),
32    path('jobseeker/education/create',
33         views.createEducation, name='createEducation'),
34    path('jobseeker/education/delete/<str:id>',
35         views.deleteEducation, name='deleteEducation'),
36    path('jobseeker/education/update/<str:id>')
```

Settings:



Results:

All clear, no errors found

views.py



CI Python Linter

```
341
342
343 @api_view(["DELETE"])
344 def deleteApplicationEmployee(request, id):
345     if request.user.is_authenticated:
346         job_post = get_object_or_404(JobPost, id=id)
347
348         # Find the application
349         try:
350             application = Application.objects.get(
351                 profile_id=request.user, job_post=job_post)
352         except Application.DoesNotExist:
353             return Response({"Error": "Application does not exist"},
354                             status=status.HTTP_404_NOT_FOUND)
355
356         # Delete the application
357         application.delete()
358
359         return Response(
360             {"Message": "success"}, status=status.HTTP_200_OK)
361     else:
362         return Response(
363             {
364                 "Error": "You are not not authorized ",
365                 status=status.HTTP_401_UNAUTHORIZED)
366
367
368 @api_view(["POST"])
369 def createUser(request):
370     serializer = CustomUserSerializer(data=request.data)
371     if serializer.is_valid():
372         serializer.save()
373         return Response(serializer.data, status=status.HTTP_200_OK)
374     else:
375         return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
376
```

Settings:



Results:

All clear, no errors found

serializers.py



CI Python Linter

```
127         'is_active',
128         'is_staff',
129         'password'
130     ]
131
132     def create(self, validated_data):
133         validated_data['password'] = make_password(validated_data['password'])
134         return super().create(validated_data)
135
136
137 class SimplifiedJobPostSerializer(serializers.ModelSerializer):
138     expiration_date = serializers.DateField(
139         format='%Y-%m-%d', input_formats=['%Y-%m-%d'])
140
141     class Meta:
142         model = JobPost
143         fields = [
144             'id',
145             'job_post_title',
146             'company_name',
147             'location',
148             'employment_type',
149             'job_description',
150             'phone_number',
151             'expiration_date',
152             'is_published'
153         ]
154
155
156 class ApplicationsSerializer(serializers.ModelSerializer):
157     job_post = SimplifiedJobPostSerializer(read_only=True)
158
159     class Meta:
160         model = Application
161         fields = ['id', 'job_post']
162
```

Settings:



Results:

All clear, no errors found

models.py



CI Python Linter

```
78     application_date = models.DateTimeField(auto_now_add=True)
79
80
81 class JobSeekerCv(models.Model):
82     profile = models.ForeignKey(
83         settings.AUTH_USER_MODEL,
84         on_delete=models.CASCADE,
85         related_name='job_seeker_profile')
86     email = models.EmailField(unique=True)
87     mobile_number = models.CharField(max_length=30)
88     applied_profiles = models.ManyToManyField(
89         JobPost, related_name='applicants')
90
91
92 class WorkExperience(models.Model):
93     job_seeker = models.ForeignKey(
94         JobSeekerCv,
95         on_delete=models.CASCADE,
96         related_name='work_experiences')
97     occupation_title = models.CharField(max_length=50)
98     company_name = models.CharField(max_length=50)
99     years = models.CharField(max_length=10)
100     description = models.CharField(max_length=500)
101
102
103 class Education(models.Model):
104     job_seeker = models.ForeignKey(
105         JobSeekerCv,
106         on_delete=models.CASCADE,
107         related_name='educations')
108     school_name = models.CharField(max_length=50)
109     level = models.CharField(max_length=50)
110     orientation = models.CharField(max_length=50)
111     description = models.CharField(max_length=500)
112     years = models.CharField(max_length=10, null=True)
113
```

Settings:



Results:

All clear, no errors found

admin.py

pep8ci.herokuapp.com/



CI Python Linter

```
1 from django.contrib import admin
2 from django.contrib.auth.admin import UserAdmin
3 from .models import CustomUser
4
5
6 class CustomUserAdmin(UserAdmin):
7     list_display = (
8         'email',
9         'first_name',
10        'last_name',
11        'is_staff',
12        'org_number',
13        'is_ag')
14     ordering = ('email',)
15     fieldsets = (
16         (None, {
17             'fields': (
18                 'email', 'password'),
19             'fields': (
20                 'first_name', 'last_name', 'org_number', 'is_ag'),
21             ('Permissions', {
22                 'fields': (
23                     'is_staff', 'is_active'))}),
24         add_fieldsets = (
25             (None, {
26                 'classes': ('wide',),
27                 'fields': ('email', 'password1', 'password2'),
28             )),
29         )
30
31
32 admin.site.register(CustomUser, CustomUserAdmin)
33
```

Settings:



Results:

All clear, no errors found