




## CHARITY

### Admin

 CI Python Linter

```
1 from django.contrib import admin
2 from .models import CharityProfile
3
4
5 admin.site.register(CharityProfile)
6
```

Settings:


 ☒ 

---

Results:

All clear, no errors found

### Forms

 CI Python Linter

```
1 from .models import CharityProfile
2 from django import forms
3
4
5 class CharityForm(forms.ModelForm):
6     class Meta:
7         model = CharityProfile
8         fields = ('charity_name', 'charity_description',
9                 'charities_coordinators')
10
```

Settings:

 ☒ 

---

Results:

All clear, no errors found


## Models



CI Python Linter

```
1 from django.db import models
2 from coordinator.models import CoordinatorProfile
3
4
5 class CharityProfile(models.Model):
6     """Details of the charity and link to which coordinators
7     they are associated with"""
8     charity_name = models.CharField(
9         max_length=50,
10        null=False,
11        blank=False,
12        verbose_name='Charity Name',
13        help_text='required, max length 50 characters',
14    )
15     charity_description = models.TextField(
16         null=False,
17         blank=False,
18         verbose_name='Charity Description',
19     )
20     charities_coordinators = models.ManyToManyField(
21         CoordinatorProfile,
22         verbose_name=('coordinators associated with charity'),
23         related_name='CharityProfiles',
24         help_text='Hold control and click to select multiple',
25     )
26
27     def __str__(self):
28         return self.charity_name
29
```


Settings:

Results:

All clear, no errors found




## Urls



CI Python Linter

```
1 from . import views
2 from django.urls import path
3
4
5 urlpatterns = [
6     path("", views.home, name="home"),
7     path("index", views.home, name="index"),
8     path("charity/add_charity", views.add_charity,
9         name="add_charity"),
10    path("charity/choose_charity", views.search_charity,
11        name="search_charity"),
12    path("charity/update_charity/cid", views.edit_charity,
13        name="edit_charity"),
14    path("charity/read_charity/cid", views.read_charity,
15        name="read_charity"),
16    path("charity/delete_charity/cid", views.delete_charity,
17        name="delete_charity"),
18 ]
19
```

Settings:

Results:

All clear, no errors found

## Views

 CI Python Linter

```
79 return render(request, 'charity/read_charity.html', context)
80
81
82 def edit_charity(request, id):
83     role = role.authenticate(request)
84     ch_profile = id
85     profile = get_object_or_404(CharityProfile, id=ch_profile)
86     if request.method == "POST":
87         form = CharityForm(request.POST, instance=profile)
88         if form.is_valid():
89             charity = form.save(commit=False)
90             form.save()
91             charity.save()
92             messages.add_message(request, messages.SUCCESS,
93                                 'Charity information updated!')
94             return redirect('dashboard')
95     form = CharityForm(instance=profile)
96     context = {
97         'form': form,
98         'profile': profile,
99         'role': role,
100     }
101     return render(request, 'charity/update_charity.html', context)
102
103
104 def delete_charity(request, id):
105     role = role.authenticate(request)
106     if role == 2:
107         charity = get_object_or_404(CharityProfile, id=id)
108         charity.delete()
109         messages.add_message(request, messages.WARNING,
110                             'Charity information deleted!')
111         return redirect('dashboard')
112     else:
113         return redirect('index')
```

**Settings:**  
🌙 ☒ 

**Results:**  
All clear, no errors found

## COORDINATOR

### Admin

 CI Python Linter

```
1 from django.contrib import admin
2 from .models import CoordinatorProfile
3
4
5 admin.site.register(CoordinatorProfile)
6
```

**Settings:**  
🌙 ☒ 

**Results:**  
All clear, no errors found

## Forms

 CI Python Linter

```
1 from django import forms
2 from .models import CoordinatorProfile
3 from volunteer.models import VolunteerProfile
4
5
6 class ProfileFormCo(forms.ModelForm):
7     class Meta:
8         model = CoordinatorProfile
9         fields = ('fname', 'lname',)
10
11
12 class ProfileFormCoUpdate(forms.ModelForm):
13     class Meta:
14         model = CoordinatorProfile
15         fields = ('fname', 'lname', 'activated',)
16
17
18 class ProfileFormVolunteer(forms.ModelForm):
19     class Meta:
20         model = VolunteerProfile
21         fields = ('activated',)
22
```

Settings:

🌙 ☒ 

---

Results:

All clear, no errors found

## Models

 CI Python Linter

```
1 from django.db import models
2 from django.contrib.auth.models import User
3
4
5 class CoordinatorProfile(models.Model):
6     """Details of the coordinator"""
7     user_name = models.OneToOneField(User, on_delete=models.CASCADE)
8     fname = models.CharField(
9         max_length=50,
10        null=False,
11        blank=False,
12        verbose_name='First Name',
13        help_text='required, max length 50 characters',
14    )
15     lname = models.CharField(
16         max_length=50,
17         null=False,
18         blank=False,
19         verbose_name='Last Name',
20         help_text='required, max length 50 characters',
21     )
22     activated = models.BooleanField(
23         default=False,
24     )
25
26
27     def __str__(self):
28         return str(self.fname + " " + self.lname)
29
```

Settings:

🌙 ☒ 

---

Results:

All clear, no errors found

## Urls



# CI Python Linter

```
1 from . import views
2 from django.urls import path
3
4
5 urlpatterns = [
6     path("", views.home, name="home"),
7     path("index", views.home, name="index"),
8     path("coordinator/pending", views.pending, name="pending"),
9     path("coordinator/dashboard", views.dashboard, name="dashboard"),
10    path("coordinator/add_profile", views.add_profile_co, name="addco"),
11    path("coordinator/update_profile/<id>", views.edit_profile_co,
12         name="updateco"),
13    path("coordinator/choose_profile", views.search_coordinators,
14         name="searchco"),
15    path("coordinator/see_profile/<id>", views.read_coordinator,
16         name="readco"),
17    path("coordinator/delete_profile/<id>", views.delete_profile_co,
18         name="deleteco"),
19    path("coordinator/search_volunteer", views.search_volunteer,
20         name="searchvol"),
21    path("coordinator/activate_volunteer/<id>", views.activate_volunteer,
22         name="activatevol"),
23    path("coordinator/activate_volunteers", views.activate_volunteers,
24         name="activatevolis"),
25 ]
26
```

Settings:

🌙 ☒ 

---

Results:

All clear, no errors found

## Views



# CI Python Linter

```
208     'volunteers_for_activation': activation,
209     'role': role,
210 }
211 return render(request, 'coordinator/activate_volunteers.html', context)
212
213
214 def activate_volunteer(request, id):
215     role = role.authenticate(request)
216     vol_id = id
217     profile = get_object_or_404(VolunteerProfile, id=vol_id)
218     skills = profile.skilled.values()
219     peoples = VolunteerProfile.objects.filter(id=vol_id).values()
220     true_pairs = [(key for key, value in people.items() if value is True)
221                  for people in peoples]
222     sessions = []
223     for true_pair in true_pairs:
224         for pair in true_pair:
225             pair = get_verbose_name(pair)
226             sessions.append(pair)
227     if request.method == "POST":
228         form = ProfileFormVolunteer(request.POST, instance=profile)
229         if form.is_valid():
230             volunteer = form.save(commit=False)
231             form.save()
232             volunteer.save()
233             messages.add_message(request, messages.SUCCESS, 'Volunteer activated!')
234             return redirect('activatevolis')
235     form = ProfileFormVolunteer(instance=profile)
236     context = {
237         'form': form,
238         'profile': profile,
239         'skills': skills,
240         'sessions': sessions,
241         'role': role
242     }
243
```

Settings:

🌙 ☒ 


---

Results:

All clear, no errors found



## ROLE

Admin

 CI Python Linter

```
1 from django.contrib import admin
2 from .models import Role, RoleChoices
3
4
5 admin.site.register(Role)
6 admin.site.register(RoleChoices)
7
```

Settings:


 ☒ 

---

Results:



All clear, no errors found

Forms

 CI Python Linter

```
1 from .models import Role
2 from django import forms
3
4
5 class RoleForm(forms.ModelForm):
6     class Meta:
7         model = Role
8         fields = ('role',)
9
```

Settings:

 ☐ 

---

Results:

All clear, no errors found

## Models

 CI Python Linter

```
1 from django.db import models
2 from django.contrib.auth.models import User
3 from django.conf import settings
4
5
6 class RoleChoices(models.Model):
7     """List of the role choices model"""
8     ROLE_CHOICES = (
9         (1, 'Volunteer'),
10        (2, 'Coordinator'),
11    )
12
13
14 class Role(models.Model):
15     """Which role and functions the user has"""
16     user_name = models.OneToOneField(User, on_delete=models.CASCADE)
17     name = models.CharField(max_length=128)
18     role = models.IntegerField(choices=RoleChoices.ROLE_CHOICES, default=1)
19
20     def __str__(self):
21         return self.name
22
```

Settings:

 ☒ 

---

Results:


All clear, no errors found

## Urls

 CI Python Linter

```
1 from . import views
2 from django.urls import path, include
3
4
5 urlpatterns = [
6     path("", views.home, name="home"),
7     path("index", views.home, name="index"),
8     path("role/role", views.role, name="role"),
9     path("login_success", views.login_success, name="login_success")
10 ]
11
```

Settings:

 ☒ 

---

Results:

All clear, no errors found


## Views



CI Python Linter

```
58 # sees whether they signed up as volunteer 1 or coordinator 2
59 title = Role.objects.filter(user_name_id=pk_logged_in).values()
60 # checks if they have a profile stored
61 vp = VolunteerProfile.objects.filter(user_name_id=pk_logged_in)
62 VP_exists = vp.exists()
63 # checks if they have a profile stored
64 cp = CoordinatorProfile.objects.filter(user_name_id=pk_logged_in)
65 CP_exists = cp.exists()
66 # gets profile so can check if activated
67 co = CoordinatorProfile.objects.filter(user_name_id=pk_logged_in)
68 co_profile = co.values()
69 # volunteer with profile
70 if title[0]['role'] == 1 and VP_exists:
71     # see volunteer profile
72     return redirect('read')
73 # volunteer without profile
74 elif title[0]['role'] == 1 and VP_exists is False:
75     # form to fill out volunteer profile
76     return redirect('add')
77 # coordinator without a profile
78 elif title[0]['role'] == 2 and CP_exists is False:
79     return redirect('addcp') # form to fill out coordinator profile
80 elif title[0]['role'] == 2 and CP_exists: # coordinator with a profile
81     # coordinator with a profile and activated
82     if co_profile[0]['activated']:
83         return redirect('dashboard')
84     # unactivated coordinator so no access
85     elif co_profile[0]['activated'] is False:
86         return redirect('pending')
87     else:
88         return redirect('index') # just in case
89 else:
90     return redirect('index') # just in case
91 else:
92     return redirect('role') # hasn't chosen role yet
93
```

Settings:

Results:

All clear, no errors found

## VOLUNTEER

### Admin



CI Python Linter

```
1 from django.contrib import admin
2 from .models import VolunteerProfile, SkillChoices
3
4
5 admin.site.register(VolunteerProfile)
6 admin.site.register(SkillChoices)
7
```

Settings:


  

Results:

All clear, no errors found






## Forms



CI Python Linter

```
1 from models import VolunteerProfile
2 from django import forms
3
4
5 class ProfileForm(forms.ModelForm):
6     class Meta:
7         model = VolunteerProfile
8         fields = ('fname', 'lname', 'phone', 'skilled',
9                 'special_skills_description',
10                'time_length_hours', 'time_length_days',
11                'mon_am', 'mon_pm', 'mon_ev', 'tue_am',
12                'tue_pm', 'tue_ev', 'wed_am', 'wed_pm',
13                'wed_ev', 'thu_am', 'thu_pm', 'thu_ev',
14                'fri_am', 'fri_pm', 'fri_ev', 'sat_am',
15                'sat_pm', 'sat_ev', 'sun_am', 'sun_pm',
16                'sun_ev',)
17
```

Settings:

Results:

All clear, no errors found

## Models



CI Python Linter

```
120 fri_pm = models.BooleanField(
121     default=False,
122     verbose_name="Friday afternoon",
123 )
124 fri_ev = models.BooleanField(
125     default=False,
126     verbose_name="Friday evening",
127 )
128 sat_am = models.BooleanField(
129     default=False,
130     verbose_name="Saturday morning",
131 )
132 sat_pm = models.BooleanField(
133     default=False,
134     verbose_name="Saturday afternoon",
135 )
136 sat_ev = models.BooleanField(
137     default=False,
138     verbose_name="Saturday evening",
139 )
140 sun_am = models.BooleanField(
141     default=False,
142     verbose_name="Sunday morning",
143 )
144 sun_pm = models.BooleanField(
145     default=False,
146     verbose_name="Sunday afternoon",
147 )
148 sun_ev = models.BooleanField(
149     default=False,
150     verbose_name="Sunday evening",
151 )
152
153 def __str__(self):
154     return self.name
155
```


Settings:

Results:

All clear, no errors found


## Urls



# CI Python Linter

```
1 from . import views
2 from django.urls import path
3
4
5 urlpatterns = [
6     path("", views.home, name="home"),
7     path("index", views.home, name="index"),
8     path("volunteer/add_profile", views.add_profile, name="add"),
9     path("volunteer/read_profile", views.read_profile, name="read"),
10    path("volunteer/edit_profile", views.edit_profile, name="edit"),
11    path("volunteer/delete_profile", views.delete_profile, name="delete"),
12 ]
13
```

Settings:

🌙 ☒ 

---

Results:

All clear, no errors found

## Views



# CI Python Linter

```
56 else:
57     print(form.errors)
58     return redirect('read')
59 context = {
60     'form': form,
61     'vp_exists': VP_exists,
62 }
63 return render(request, 'volunteer/add_profile.html', context)
64
65
66 def edit_profile(request):
67     pk_logged_in = request.user.pk
68     available = get_object_or_404(VolunteerProfile, user_name_id=pk_logged_in)
69     if request.method == 'POST':
70         form = ProfileForm(request.POST, instance=available)
71         if form.is_valid():
72             form.save()
73             print(form)
74             messages.add_message(request, messages.SUCCESS, 'Profile Updated!')
75             return redirect('read')
76     form = ProfileForm(instance=available)
77     context = {
78         'form': form,
79         'availabilities': available,
80         'pk_logged_in': pk_logged_in
81     }
82     return render(request, 'volunteer/edit_profile.html', context)
83
84
85 def delete_profile(request):
86     pk_logged_in = request.user.pk
87     volunteer = get_object_or_404(User, id=pk_logged_in)
88     volunteer.delete()
89     messages.add_message(request, messages.WARNING, 'Profile deleted!')
90     return redirect('index')
91
```

Settings:

🌙 ☒ 


---

Results:

All clear, no errors found

## VOLUNTEER VILLAGE




Settings – Has four errors but this is in standard code that is generated by other sources not myself so I am not going to adjust this. I have put it on the authorization section in the screenshot so the overlength lines can be seen to be not part of my work.



CI Python Linter

```
112
113 DATABASES = {
114     'default': dj_database_url.parse(os.environ.get("DATABASE_URL"))
115 }
116
117
118 # Password validation
119 # https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators
120
121 AUTH_PASSWORD_VALIDATORS = [
122     {
123         'NAME': 'django.contrib.auth.password_validation.UserAttributesSimilarityValidator',
124     },
125     {
126         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
127     },
128     {
129         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
130     },
131     {
132         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
133     },
134 ]
135
136
137 # Internationalization
138 # https://docs.djangoproject.com/en/4.2/topics/i18n/
139
140 LANGUAGE_CODE = 'en-us'
141
142 TIME_ZONE = 'UTC'
143
144 USE_I18N = True
145
146 USE_TZ = True
147
```

Settings:

Results:

123: E501 line too long (91 > 79 characters)

126: E501 line too long (81 > 79 characters)

129: E501 line too long (82 > 79 characters)

132: E501 line too long (83 > 79 characters)

## Urls



CI Python Linter

```
1  """
2  URL configuration for volunteervillage project.
3
4  The 'urlpatterns' list routes URLs to views. For more information please see:
5  https://docs.djangoproject.com/en/4.2/topics/http/urls/
6  Examples:
7  Function views
8      1. Add an import:  from my_app import views
9      2. Add a URL to urlpatterns:  path('', views.home, name='home')
10 Class-based views
11     1. Add an import:  from other_app.views import Home
12     2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path, include
19
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path("", include('role.urls'), name="role-urls"),
24     path("", include("coordinator.urls"), name="coordinator-urls"),
25     path("", include("volunteer.urls"), name="volunteer-urls"),
26     path("", include("charity.urls"), name="charity-urls"),
27     path('accounts/', include('allauth.urls')),
28 ]
29
```

Settings:

Results:

All clear, no errors found