class Profile(models.Model):  
 first\_name = models.CharField(  
 max\_length=25,  
 validators=(  
 MinLengthValidator(2),  
 ),  
 )  
  
 last\_name = models.CharField(  
 max\_length=25,  
 validators=(  
 MinLengthValidator(2),  
 ),  
 )  
  
 email = models.EmailField()  
  
 age = models.IntegerField(  
 null=True,  
 blank=True,  
 validators=(  
 MinValueValidator(0),  
 ),  
 )  
  
 nationality = models.CharField(  
 max\_length=20,  
 null=True,  
 blank=True,  
 validators=(  
 MinLengthValidator(2),  
 ),  
 )  
  
 total\_cars = models.IntegerField(  
 default=0,  
 )  
  
 total\_likes = models.IntegerField(  
 default=0,  
 )  
  
 @property  
 def full\_name(self):  
 return f'{self.first\_name} {self.last\_name}'  
  
 def \_\_str\_\_(self):  
 return self.full\_name

class CreateProfileForm(forms.ModelForm):  
 class Meta:  
 model = Profile  
 fields = '\_\_all\_\_'  
  
  
class DeleteProfileForm(forms.ModelForm):  
  
 def save(self, commit=True):  
 self.instance.delete()  
 Car.objects.all().delete()  
 return self.instance  
  
 class Meta:  
 model = Profile  
 fields = ()

def get\_profile():  
 profiles = Profile.objects.all()  
 if profiles:  
 return profiles[0]  
 return None

def create\_profile(request):  
 if request.method == 'POST':  
 form = CreateProfileForm(request.POST, request.FILES)  
 if form.is\_valid():  
 form.save()  
 return redirect('show index')  
 else:  
 form = CreateProfileForm()  
  
 context = {  
 'form': form,  
 'no\_profile': True,  
 }  
 return render(request, 'home-no-profile.html', context)

path('profile/create/', create\_profile, name='create profile'),

path('profile/delete/', delete\_profile, name='delete profile'),

def delete\_profile(request):  
 profile = get\_profile()  
 if request.method == 'POST':  
 form = DeleteProfileForm(request.POST, request.FILES, instance=profile)  
 if form.is\_valid():  
 form.save()  
 return redirect('show index')  
 else:  
 form = DeleteProfileForm(instance=profile)  
  
 context = {  
 'form': form,  
 }  
 return render(request, 'profile-delete.html', context)

def profile\_details(request):  
 profile = get\_profile()  
  
 cars = Car.objects.all()  
  
 context = {  
 'profile': profile,  
 'cars\_count': len(cars),  
 }  
 return render(request, 'profile-details.html', context)

path('profile/details/', profile\_details, name='profile details'),

# def test\_expect\_correct\_template(self):  
# user = UserModel.objects.create\_user(\*\*self.VALID\_USER\_CREDENTIALS)  
# profile = Profile.objects.create(  
# \*\*self.VALID\_PROFILE\_DATA,  
# user=user,  
# )  
#  
# response = self.client.get(reverse('profile details', kwargs={  
# 'pk': profile.pk,  
# }))  
# self.assertTemplateUsed('accounts/profile-details.html')