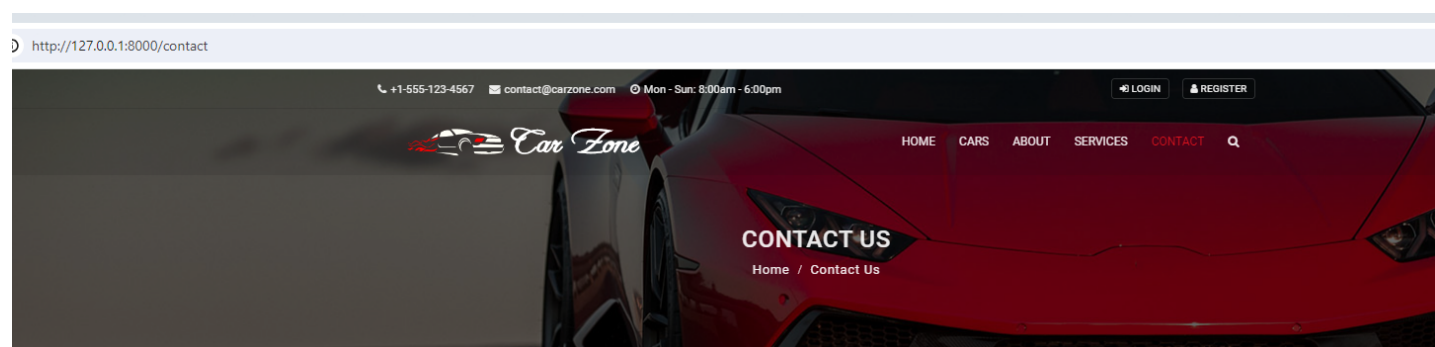


Topic: Car Listing: Contact Us Page & Send Emails Part 18

Speaker: Udemy Instructor Rathan Kumar / Notebook: Django Project: Car Listing



1. We make our CONTACT page functional. WE DONT SAVE the details of this Contact Form into our database but instead, we will send all these details as an EMAIL to our ADMIN.



Contact Us

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<div>Write message</div>	
<div>SEND MESSAGE</div>	

Phone:
0477 8556 552

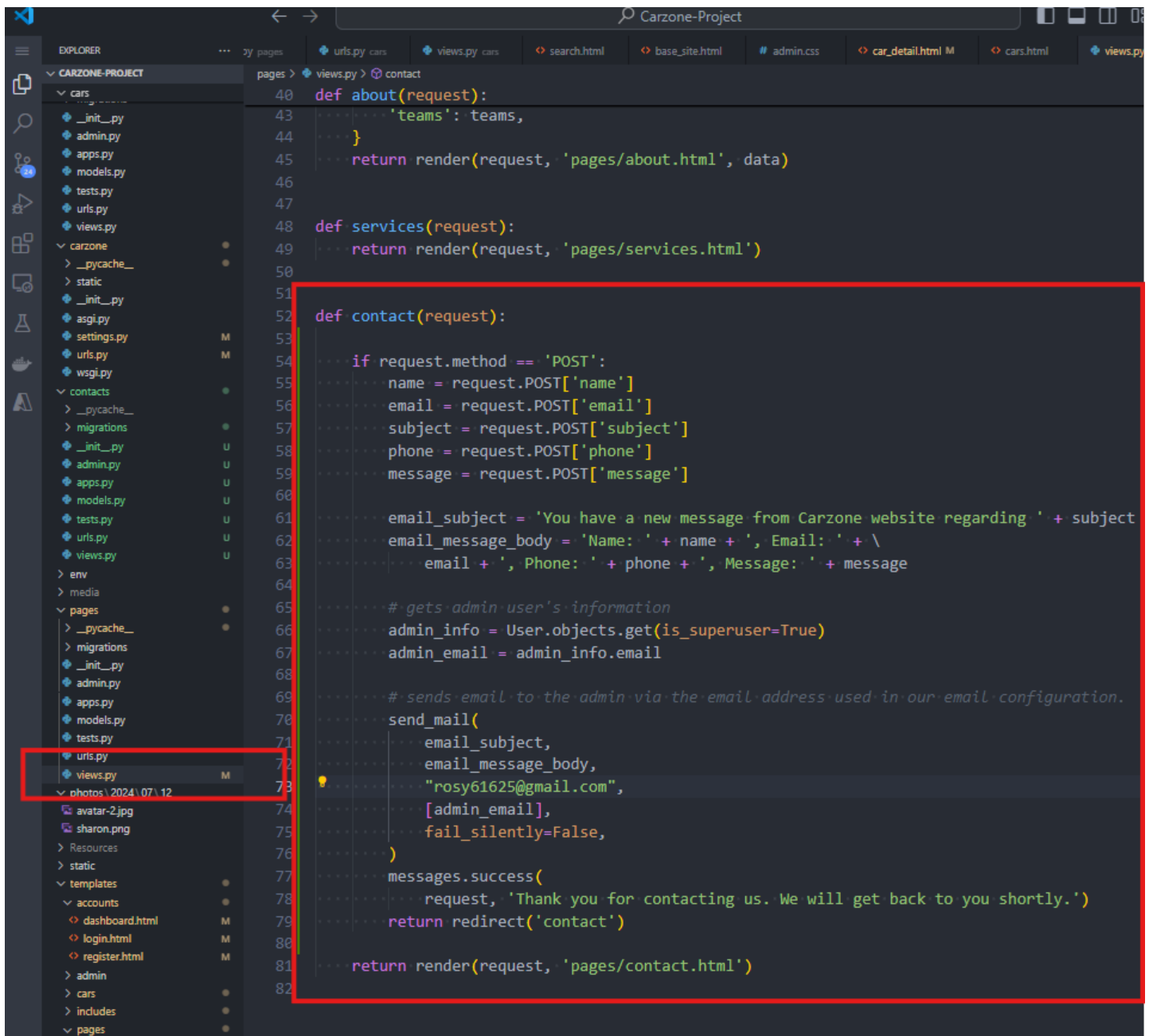
Email:
info@themevessel.com

Web:
info@themevessel.com

Fax:
+0477 85X6 552

2. Update the PAGES\VIEWS.PY

```
1 from django.shortcuts import render, redirect
2 from .models import Team
3 from cars.models import Car
4
5 from django.core.mail import send_mail
6 from django.contrib.auth.models import User
7 from django.contrib import messages
8
9
10 # Create your views here.
11 def home(request):
12     """# get all the team members"""
13     teams = Team.objects.all()
14     featured_cars = Car.objects.order_by(
15         """# '-created_date'""").filter(is_featured=True)
16     all_cars = Car.objects.order_by("""# '-created_date'""")
17     """# extracts the values of these fields for searching"""
18     """# search_fields = Car.objects.values('model', 'city', 'year', 'body_style')"""
19
20     """# extracts only unique values on models into a list"""
21     model_search = Car.objects.values_list('model', flat=True).distinct()
22     city_search = Car.objects.values_list('city', flat=True).distinct()
23     year_search = Car.objects.values_list('year', flat=True).distinct()
24     body_style_search = Car.objects.values_list(
25         """# 'body_style'""", flat=True).distinct()
26     data = {
27         """# 'teams'""": teams,
28         """# 'featured_cars'""": featured_cars,
29         """# 'all_cars'""": all_cars,
30         """# 'search_fields'""": search_fields,
31         """# 'model_search'""": model_search,
32         """# 'city_search'""": city_search,
33         """# 'year_search'""": year_search,
```



```
40 def about(request):
41     ...
42     ... 'teams': teams,
43     ...
44 }
45 return render(request, 'pages/about.html', data)
46
47
48 def services(request):
49     return render(request, 'pages/services.html')
50
51
52 def contact(request):
53     ...
54     if request.method == 'POST':
55         name = request.POST['name']
56         email = request.POST['email']
57         subject = request.POST['subject']
58         phone = request.POST['phone']
59         message = request.POST['message']
60
61         email_subject = 'You have a new message from Carzone website regarding ' + subject
62         email_message_body = 'Name: ' + name + ', Email: ' + \
63             email + ', Phone: ' + phone + ', Message: ' + message
64
65         # gets admin user's information
66         admin_info = User.objects.get(is_superuser=True)
67         admin_email = admin_info.email
68
69         # sends email to the admin via the email address used in our email configuration.
70         send_mail(
71             email_subject,
72             email_message_body,
73             "rosy61625@gmail.com",
74             [admin_email],
75             fail_silently=False,
76         )
77         messages.success(
78             request, 'Thank you for contacting us. We will get back to you shortly.')
79         return redirect('contact')
80
81     return render(request, 'pages/contact.html')
82
```

3. Update the CONTACT.HTML as:

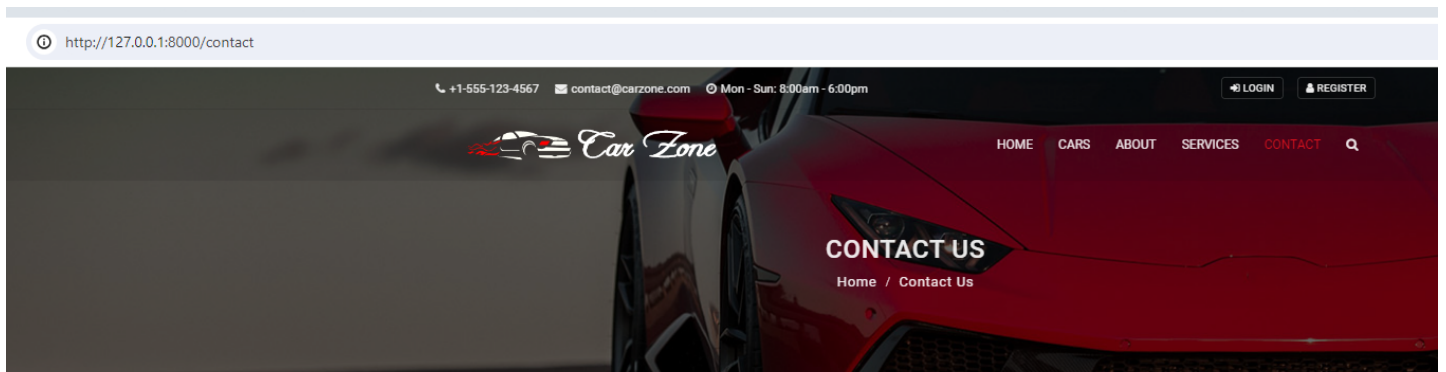
The screenshot shows a VS Code editor with the 'Carzone-Project' open. The Explorer sidebar on the left shows the project structure, with 'contact.html' highlighted under the 'pages' directory. The main editor displays the content of 'contact.html', which is a Django template. A red box highlights a section of the template code, including a form action, CSRF token, and an include statement for a message template.

```

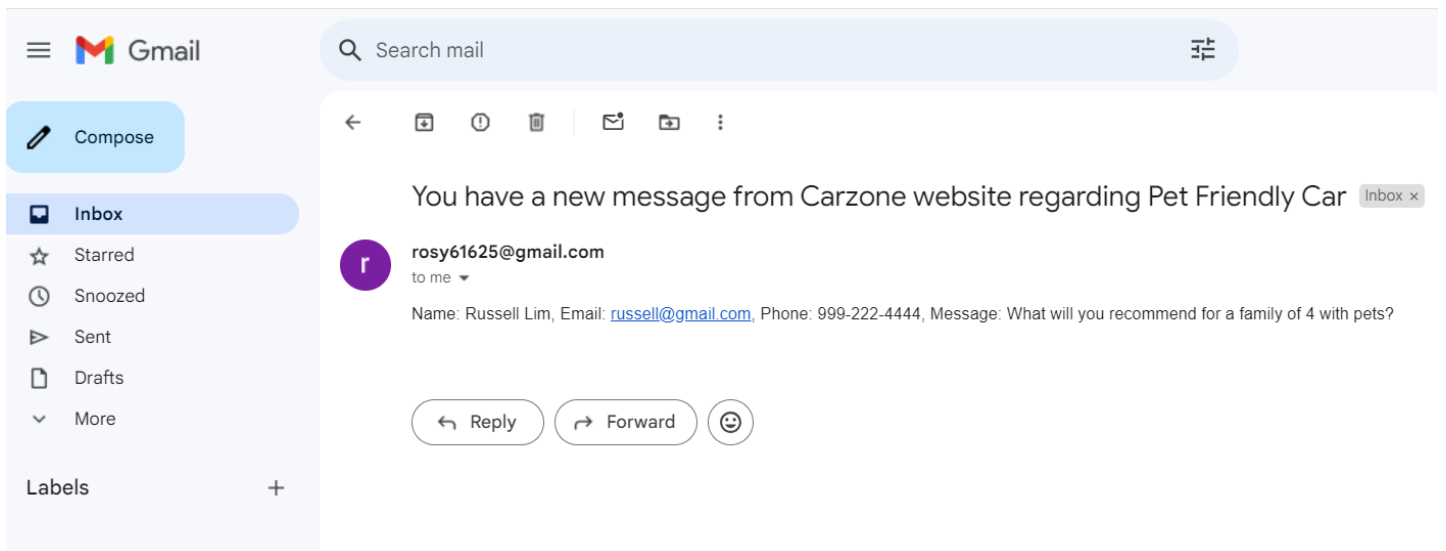
21
22 <!-- Contact 2 start -->
23 <div class="contact-2 content-area-5">
24 <!--<div class="container">
25 <!--<!-- Main title -->
26 <!--<div class="main-title text-center">
27
28
29 <!--<h1>Contact <span>Us</span></h1>
30 <!--<p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod</p>
31 <!--</div>
32 <!--<form action="{% url 'contact' %}" method="POST" enctype="multipart/form-data">
33 <!--<{% csrf_token %}
34
35 <!--<!-- inserts the customized message -->
36 <!--<{% include 'includes/messages.html' %}
37
38 <!--<div class="row">
39 <!--<div class="col-lg-7">
40 <!--<div class="row">
41 <!--<div class="col-md-6 text-left">
42 <!--<div class="form-group name">
43 <!--<input type="text" name="name" class="form-control" placeholder="Name" value="">
44 <!--</div>
45 <!--</div>
46 <!--<div class="col-md-6">
47 <!--<div class="form-group email">
48 <!--<input type="email" name="email" class="form-control" placeholder="Email" value="">
49 <!--</div>
50 <!--</div>
51 <!--<div class="col-md-6">
52 <!--<div class="form-group subject">
53 <!--<input type="text" name="subject" class="form-control" placeholder="Subject" value="">
54 <!--</div>
55 <!--</div>
56 <!--<div class="col-md-6">
57 <!--<div class="form-group number">
58 <!--<input type="text" name="phone" class="form-control" placeholder="Phone" value="">
59 <!--</div>
60 <!--</div>
61 <!--<div class="col-md-12">
62 <!--<div class="form-group message">
63 <!--<textarea class="form-control" name="message" placeholder="Write your message here" value="">
64 <!--</div>
65 <!--</div>
66 <!--</div>

```

4. Test it.



See your email (the admin user email address). If you want to format the email, you need to use `RENDER_TO_STRING` (see your Notebook project and Blog Project for reference).



4.