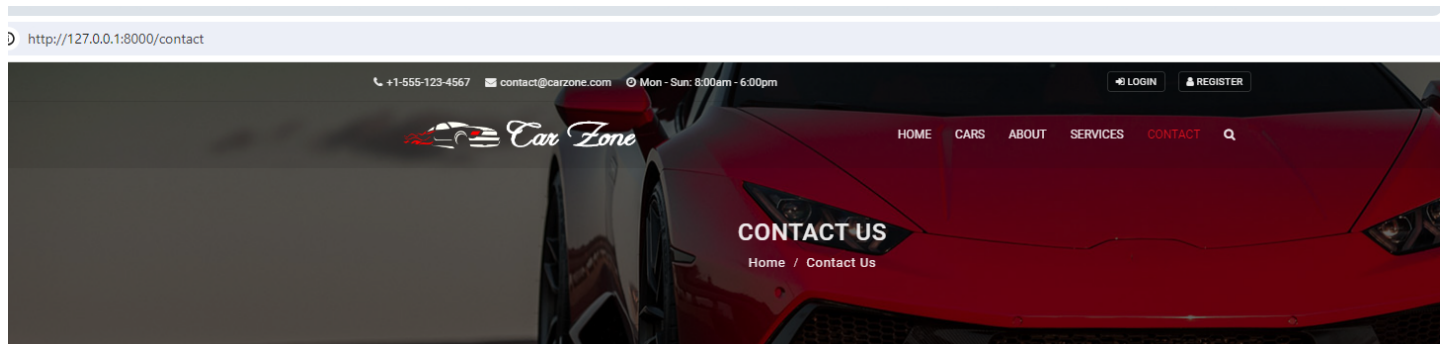


Topic: Car Listing: Contact Us Page & Send Emails Part 18

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 adipiscing elit, sed do eiusmod

<input type="text" value="Full Name"/>	<input type="text" value="Email"/>
<input type="text" value="Subject"/>	<input type="text" value="Phone Number"/>
<input type="text" value="Write message"/>	
<input type="button" value="SEND MESSAGE"/>	

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,
```

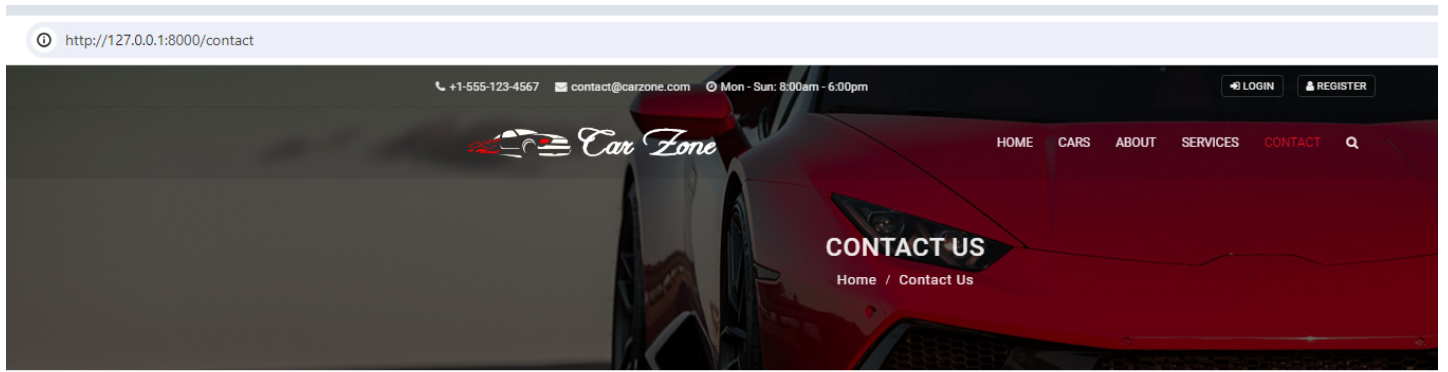
```
40 def about(request):
41     context = {'teams': teams,
42               }
43     return render(request, 'pages/about.html', data)
44
45 def services(request):
46     return render(request, 'pages/services.html')
47
48 def contact(request):
49     if request.method == 'POST':
50         name = request.POST['name']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

3. Update the CONTACT.HTML as:

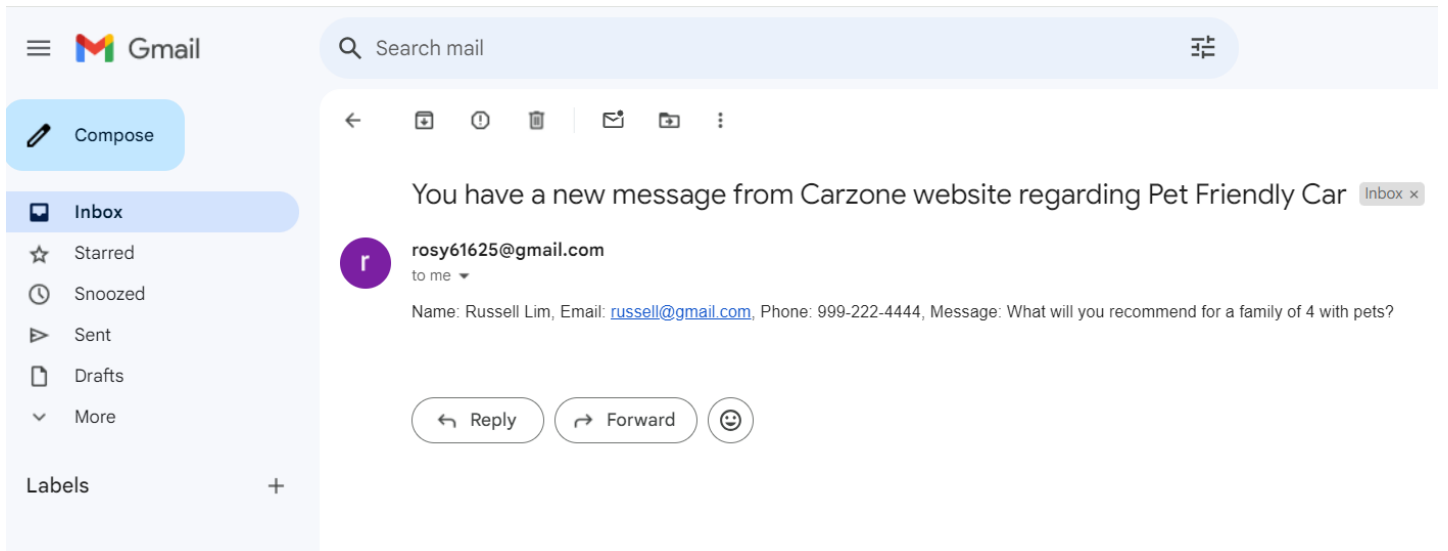
```
Carzone-Project
templates > pages > contact.html > div.contact-2.content-area-5 > div.container > form
CARZONE-PROJECT
cars
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
carzone
  __pycache__
  static
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
contacts
  __pycache__
  migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
env
media
pages
  __pycache__
  migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
photos
Resources
static
templates
  accounts
    dashboard.html
    login.html
    register.html
  admin
  cars
  includes
  pages
    account.html
    contact.html
    home.html
    services.html
  base.html
.gitignore
carzone.db
```

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