

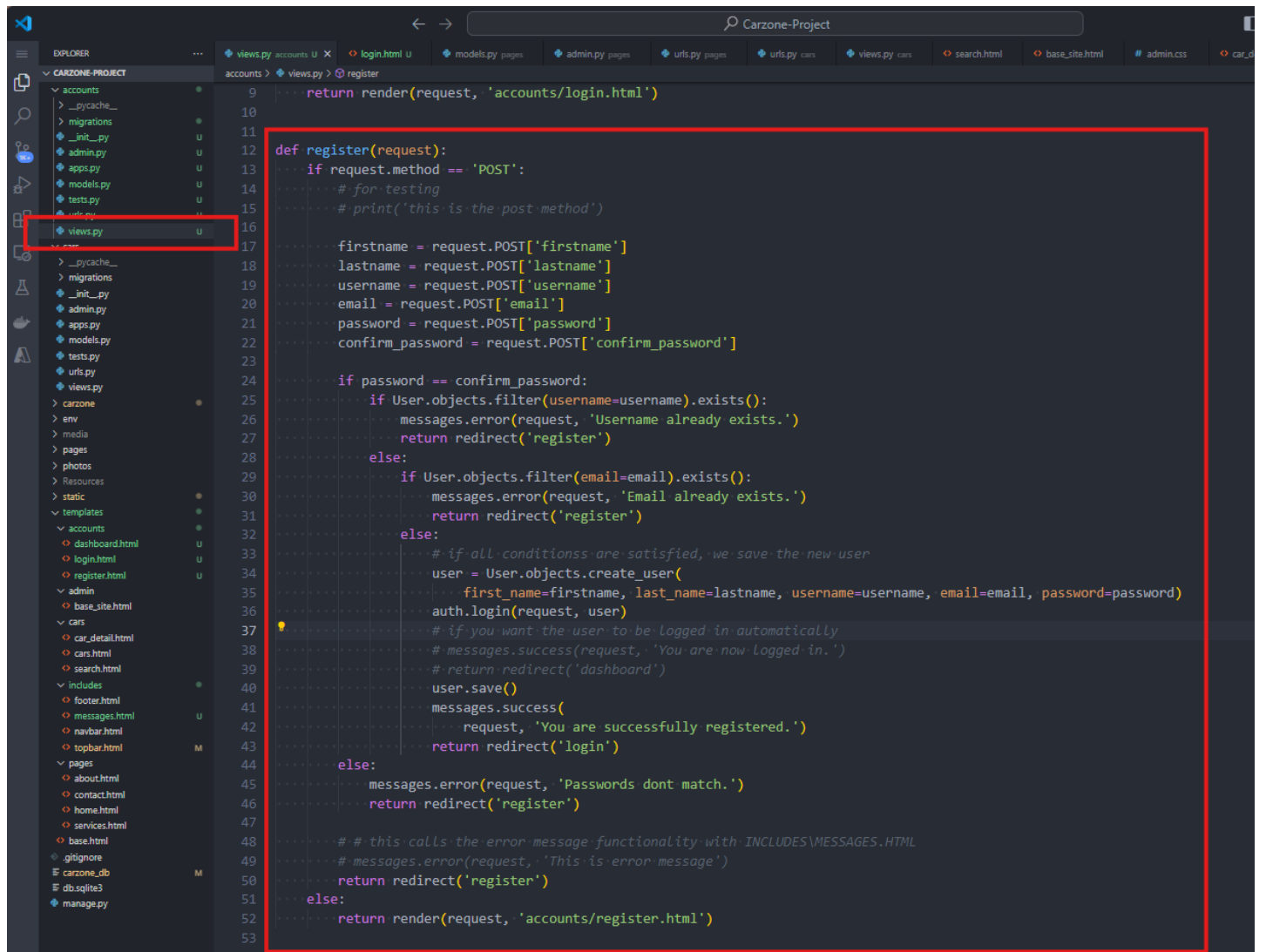
Topic: Car Listing: User Registration, Login, Logout, Dynamic Page Title Part 13

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



This is the continuation of the REGISTER FUNCTIONALITY from Part 12.

1. Go to ACCOUNTS/VIEWS.PY and update



```
9         return render(request, 'accounts/login.html')
10
11
12 def register(request):
13     if request.method == 'POST':
14         # for testing
15         # print('this is the post method')
16
17         # username = request.POST['username']
18         # lastname = request.POST['lastname']
19         # email = request.POST['email']
20         # password = request.POST['password']
21         # confirm_password = request.POST['confirm_password']
22
23         # if password == confirm_password:
24             # if User.objects.filter(username=username).exists():
25                 # messages.error(request, 'Username already exists.')
26                 # return redirect('register')
27             # else:
28                 # if User.objects.filter(email=email).exists():
29                     # messages.error(request, 'Email already exists.')
30                     # return redirect('register')
31                 # else:
32                     # if all conditionss are satisfied, we save the new user
33                     # user = User.objects.create_user(
34                         # first_name=firstname, last_name=lastname, username=username, email=email, password=password)
35                     # auth.login(request, user)
36                     # if you want the user to be logged in automatically
37                     # messages.success(request, 'You are now Logged in.')
38                     # return redirect('dashboard')
39                     # user.save()
40                     # messages.success(
41                         # request, 'You are successfully registered.')
42                     # return redirect('login')
43             # else:
44                 # messages.error(request, 'Passwords dont match.')
45                 # return redirect('register')
46
47     # # this calls the error message functionality with INCLUDES/MESSAGES.HTML
48     # messages.error(request, 'This is error message')
49     # return redirect('register')
50 # else:
51     # return render(request, 'accounts/register.html')
52
53
54
```

2. Update your REGISTER.HTML

```
21 <!-- Contact section start -->
22 </div class="contact-section">
23 <div class="container">
24 <div class="row">
25 <div class="col-lg-12 col-md-12">
26 <div class="form-section">
27 <div class="logo-2">
28 <a href="{% url 'home' %}">
29 
30 </a>
31 </div>
32 <h3>Create an account</h3>
33
34 <!-- This displays our customized error message -->
35 <{% include 'includes/messages.html' %}>
36
37 <form action="{% url 'register' %}" method="POST">
38 <!-- this is for security reason that we add the CSRF_TOKEN -->
39 <{% csrf_token %}>
40 <div class="form-group form-box">
41 <input type="text" name="firstname" class="input-text" placeholder="First Name" required>
42 <i class="fa fa-user"></i>
43 </div>
44 <div class="form-group form-box">
45 <input type="text" name="lastname" class="input-text" placeholder="Last Name" required>
46 <i class="fa fa-user"></i>
47 </div>
48 <div class="form-group form-box">
49 <input type="text" name="username" class="input-text" placeholder="Username" required>
50 <i class="fa fa-user"></i>
51 </div>
52 <div class="form-group form-box">
53 <input type="email" name="email" class="input-text" placeholder="Email Address" required>
54 <i class="flaticon-mail"></i>
55 </div>
56 <div class="form-group form-box">
57 <input type="password" name="password" class="input-text" placeholder="Password" required>
58 <i class="flaticon-lock"></i>
59 </div>
60 <div class="form-group form-box">
61 <input type="password" name="confirm_password" class="input-text"
62 placeholder="Confirm Password" required>
63 <i class="flaticon-lock"></i>
64 </div>
65 <div class="form-group mb-0 clearfix">
66 <button type="submit" class="btn-md btn-theme float-left">Register</button>
67 </div>
```

3. You can test it. We updated our VIEWS.PY if we want the user to be logged in automatically or not.

4. Update the LOGIN.HTML as

FROM:

```
<h3>Sign into your account</h3>
<form action="{% url 'login' %}" method="POST">

    {% csrf_token %}
    <div class="form-group form-box">
        <input type="email" name="email" class="input-text" placeholder="Email Address" required>
        <i class="flaticon-mail"></i>
    </div>
    <div class="form-group form-box">
        <input type="password" name="password" class="input-text" placeholder="Password" required>
        <i class="flaticon-lock"></i>
    </div>
    <div class="form-group mb-0 clearfix">
        <button type="submit" class="btn-md btn-theme float-left">Login</button>
    </div>
    <div class="extra-login clearfix">
        <span>Or Login With</span>
    </div>
    <div class="clearfix"></div>
    <ul class="social-list">
        <li><a href="#" class="facebook-bg"><i
            class="fa fa-facebook facebook-i"></i><span>Facebook</span></a></li>
        <li><a href="#" class="google-bg"><i
            class="fa fa-google google-i"></i><span>Google</span></a></li>
    </ul>
</form>
```

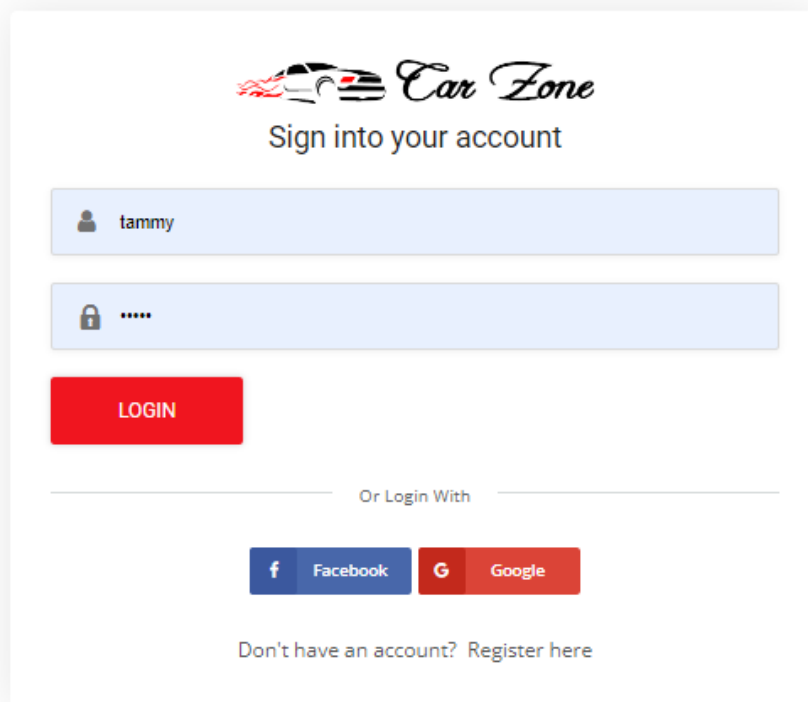
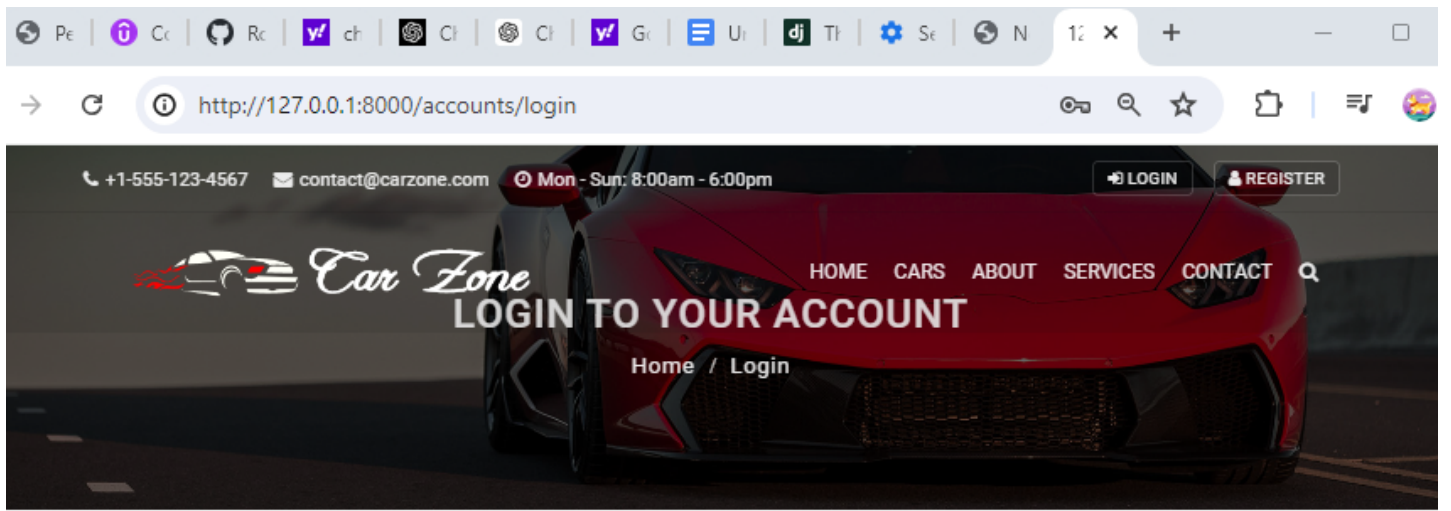
TO:

```
20 <div class="contact-section">
21     <div class="container">
22         <div class="row">
23             <div class="col-lg-12">
24                 <div class="form-section">
25                     <a href="{% url 'home' %}">
26                         
27                     </a>
28                 </div>
29             </div>
30             <h3>Sign into your account</h3>
31             <!-- inserts the customized error messages -->
32             {% include 'includes/messages.html' %}
33         </div>
34         <form action="{% url 'login' %}" method="POST">
35             {% csrf_token %}
36             <div class="form-group form-box">
37                 <input type="text" name="username" class="input-text" placeholder="username" required>
38                 <i class="fa fa-user"></i>
39             </div>
40             <div class="form-group form-box">
41                 <input type="password" name="password" class="input-text" placeholder="Password" required>
42                 <i class="flaticon-lock"></i>
43             </div>
44             <div class="form-group mb-0 clearfix">
45                 <button type="submit" class="btn-md btn-theme float-left">Login</button>
46             </div>
47             <div class="extra-login clearfix">
48                 <span>Or Login With</span>
49             </div>
50             <div class="clearfix"></div>
51             <ul class="social-list">
52                 <li><a href="#" class="facebook-bg"><i
53                     class="fa fa-facebook facebook-i"></i><span>Facebook</span></a></li>
54                 <li><a href="#" class="google-bg"><i
55                     class="fa fa-google google-i"></i><span>Google</span></a></li>
56             </ul>
57         </form>
58     </div>
```

5. Now, for the LOGIN function, update our VIEWS.PY as

```
1 from django.shortcuts import render, redirect
2 from django.contrib import messages, auth
3 from django.contrib.auth.models import User
4
5 # Create your views here.
6
7
8 def login(request):
9
10     if request.method == 'POST':
11         # extracts the user form's input
12         username = request.POST['username']
13         password = request.POST['password']
14
15         # checks the user credentials with our db record
16         user = auth.authenticate(username=username, password=password)
17
18         if user is not None:
19             auth.login(request, user)
20             messages.success(request, 'You are now logged in.')
21             return redirect('dashboard')
22         else:
23             messages.error(request, 'Invalid login credentials.')
24             return redirect('login')
25     return render(request, 'accounts/login.html')
26
```

6. Run the server and test the LOGIN. You should be able to see the error message if incorrect credentials were used otherwise, we should be logged in to our DASHBOARD.



7. For LOGOUT, we update our TOPBAR.HTML to show a LOGOUT if the user has logged in.

FROM:

```

<ul class="top-social-media pull-right">
  <li>
    <a href="{% url 'login' %}" class="sign-in"><i class="fa fa-sign-in"></i> Login</a>
  </li>
  <li>
    <a href="{% url 'register' %}" class="sign-in"><i class="fa fa-user"></i> Register</a>
  </li>
</ul>

```

TO:

```

<div class="col-lg-4 col-md-4 col-sm-5">
  <ul class="top-social-media pull-right">
    <!-- checks if the user has successfully logged in -->
    {% if user.is_authenticated %}
      <li>
        <a href="{% url 'dashboard' %}" class="sign-in"><i class="fa fa-tachometer"></i> Dashboard</a>
      </li>
      <li>
        <!-- There are other ways to do this logout, but this is the preferred way -->
        <a href="javascript:(document.getElementById('logout').submit())" class="sign-out"><i
          class="fa fa-sign-out"></i> Logout</a>
        <form action="{% url 'logout' %}" id="logout" method="POST">
          <!-- this form does not do show anything -->
          {% csrf_token %}
          <input type="hidden">
        </form>
      </li>
    {% else %}
      <li>
        <a href="{% url 'login' %}" class="sign-in"><i class="fa fa-sign-in"></i> Login</a>
      </li>
      <li>
        <a href="{% url 'register' %}" class="sign-in"><i class="fa fa-user"></i> Register</a>
      </li>
    {% endif %}
  </ul>

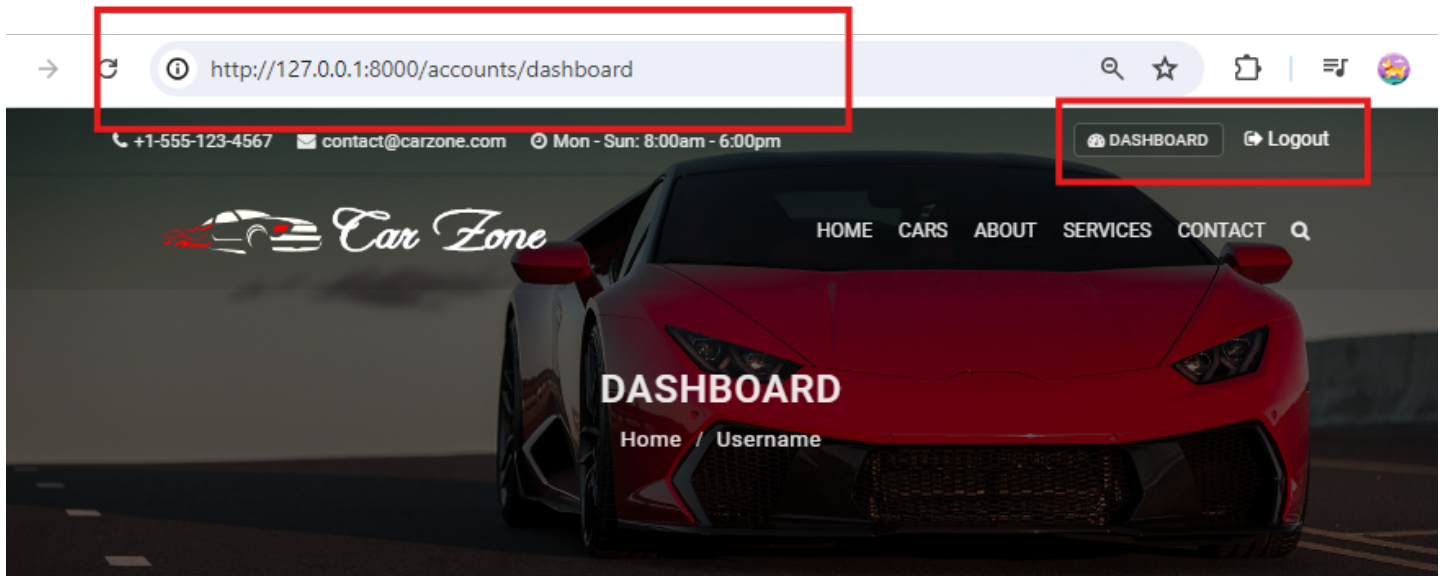
```

8. We update our VIEWS.PY to accept the FORM LOGOUT POST METHOD.

```
accounts > views.py > ...
accounts
  > __pycache__
  > migrations
  > __init__.py
  > admin.py
  > apps.py
  > models.py
  > tests.py
  > urls.py
  > views.py
carzone
  > env
  > media
  > pages
  > photos
  > Resources
  > static
  > templates
  > accounts
    <> dashboard.html
    <> login.html

28 def register(request):
29     ...
30     ... return redirect('login')
31
32     ... else:
33     ...     messages.error(request, 'Passwords dont match.')
34     ...     return redirect('register')
35
36     ... # this calls the error message functionality with INCLUDES\MESSAGES.H
37     ... # messages.error(request, 'This is error message')
38     ... return redirect('register')
39
40     ... else:
41     ...     return render(request, 'accounts/register.html')
42
43
44 def dashboard(request):
45     ... return render(request, 'accounts/dashboard.html')
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72 def dashboard(request):
73     ... return render(request, 'accounts/dashboard.html')
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176 def logout(request):
177     ... if request.method == 'POST':
178     ...     auth.logout(request)
179     ...     messages.success(request, 'You are successfully logged out.')
180     ...     return redirect('home')
181     ... return redirect('home')
```

9. When you are in the dashboard and need to logout, you should be directed to the homepage.



Welcome User

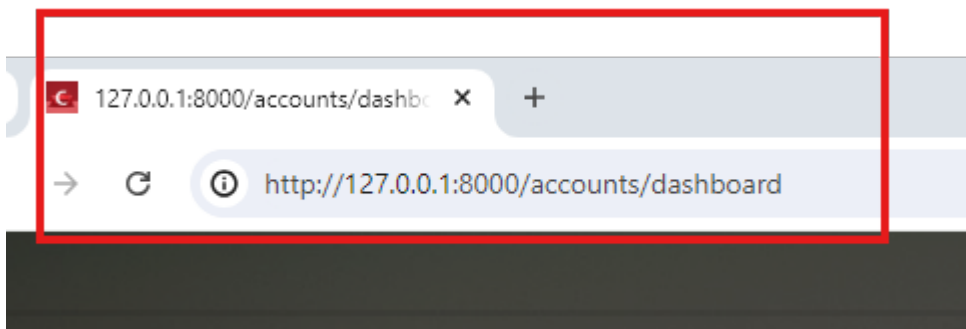
Here are the cars that you have inquired about

#	Car Name	Location	Price	Action
1	Lamborghini Huracan	Austin	\$25000	View Car
2	Lamborghini Huracan	Austin	\$25000	View Car
3	Lamborghini Huracan	Austin	\$25000	View Car

© 2024 Carzone Corp. All Rights Reserved.



10. We want to change the web page title dynamically based on what webpage we are in.



We go to our TEMPLATES\BASE.HTML and find the TITLE SECTION.

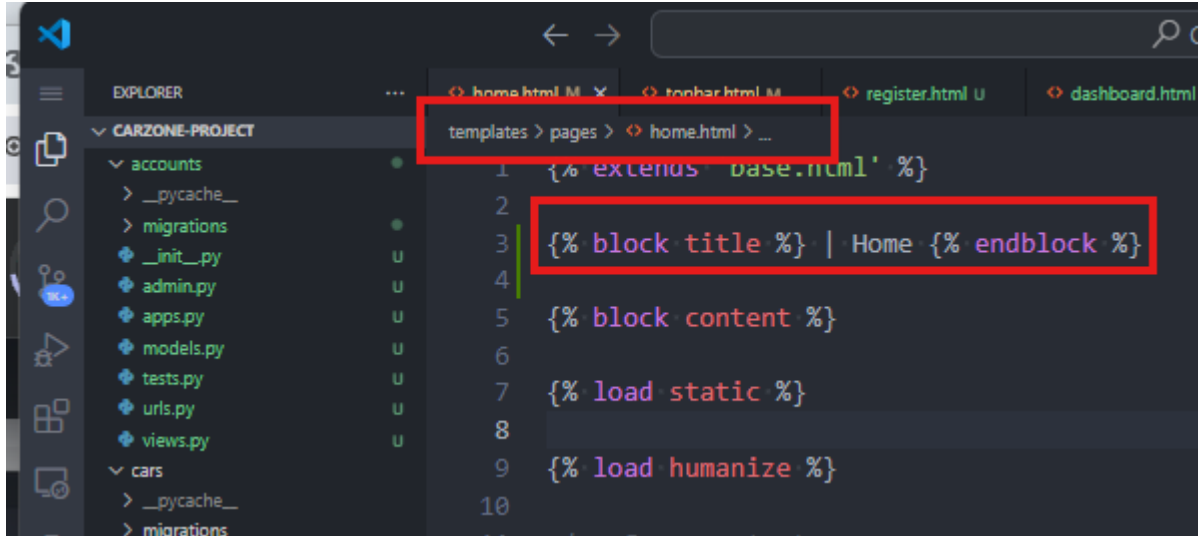
FROM:

```
<head>
...<title></title>
...<meta name="viewport" content="width=device-width, init
...<meta charset="utf-8">
```

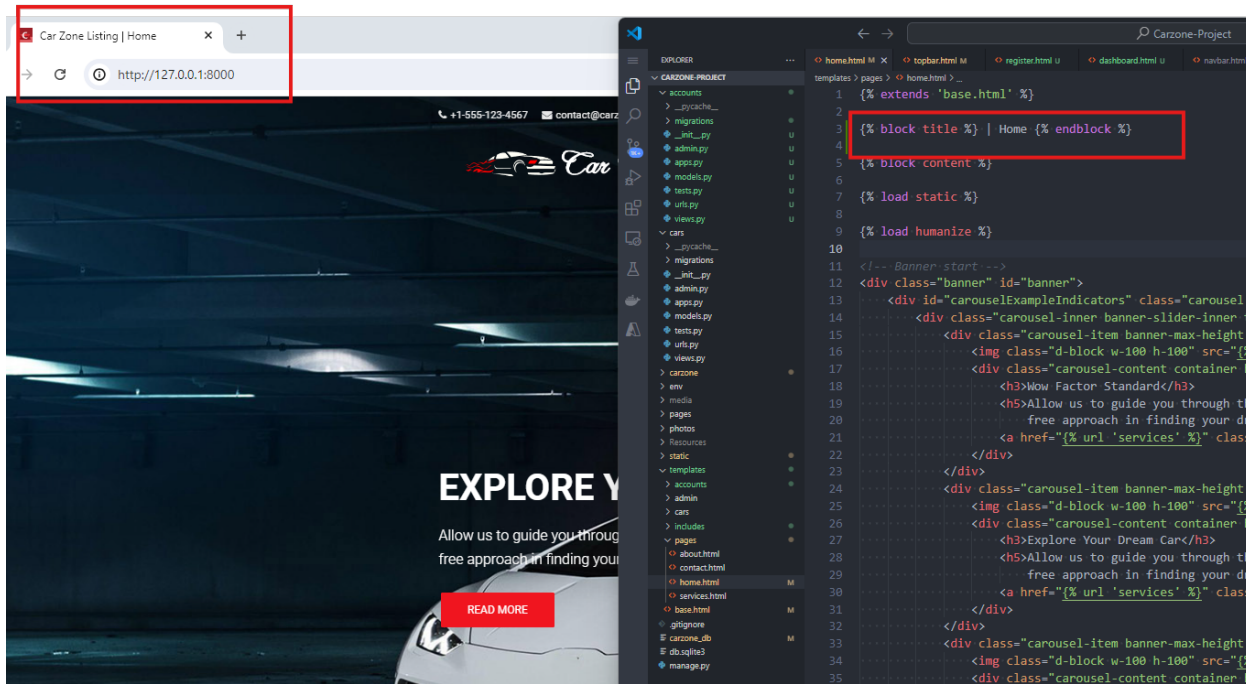

TO:

```
<head>
  <title>Car Zone Listing {% block title %} {% endblock %}</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta charset="utf-8">
```

11. Then in each webpage like PAGESHOME.HTML, update it as



So when you run your server. You need to update each of your webpages to write the customized page title.



12. To show the CAR TITLE as part of the PAGE TITLE, we modify CAR_DETAILS.HTML as:

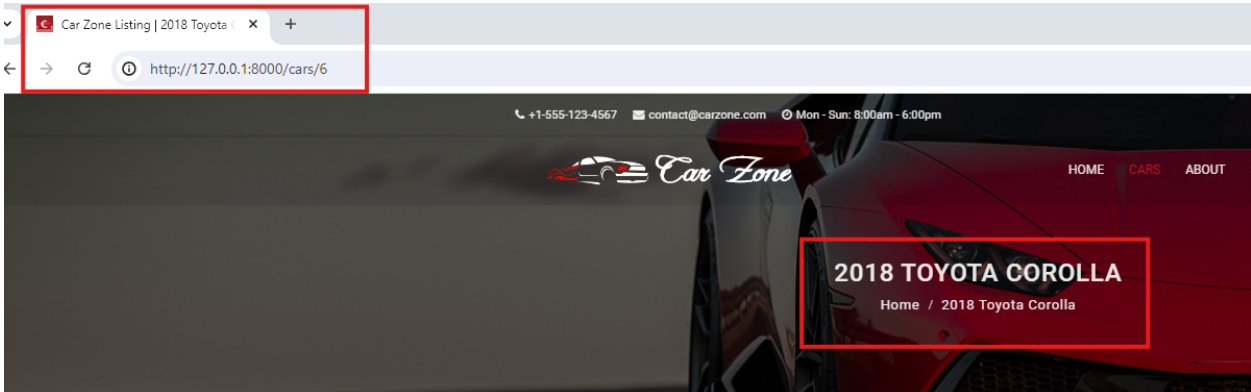
FROM:

```
1 {% extends 'base.html' %}
2
3 {% block title %} | Car Details {% endblock %}
4
5
6 {% block content %}
7
8 {% load static %}
```

TO:

```
templates > cars > car_detail.html > div.sub-banner.overview-bgi
1 {% extends 'base.html' %}
2
3 {% block title %} | {{ single_car.car_title }} {% endblock %}
4
5
6 {% block content %}
7
8 {% load static %}
9
10 {% load humanize %}
```

Click on a specific car. The car title should be visible as part of the page title.



2018 TOYOTA COROLLA

\$18,000

Indianapolis, IN



Refine Your Search

Color:

Model:

Year:

Body Style:

Engine:

Transmission:

Interior: