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

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

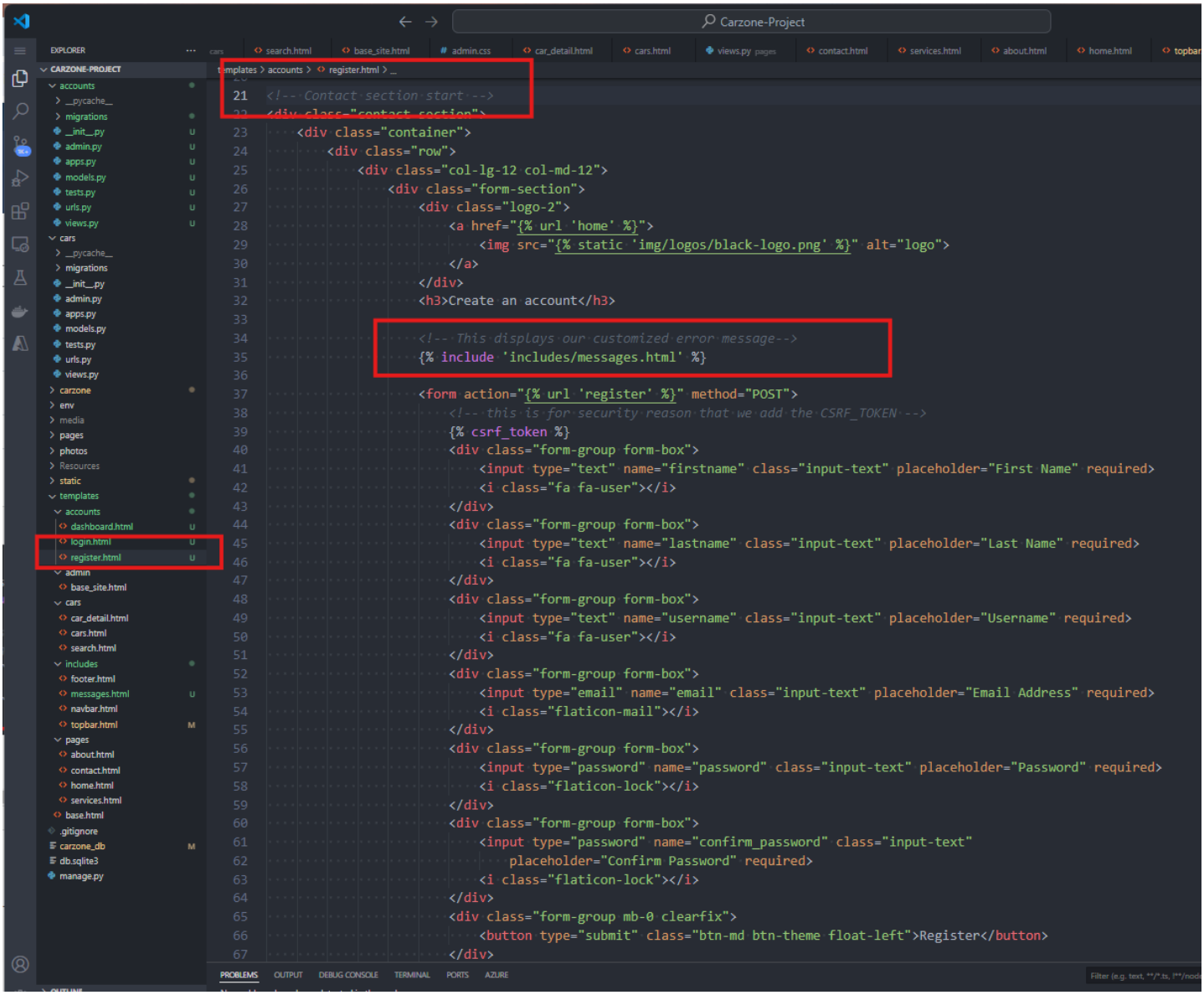
---

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

1. Go to ACCOUNTS\VIEWS.PY and update

```python
    return render(request, 'accounts/login.html')


def register(request):
    if request.method == 'POST':
        # for testing
        # print('this is the post method')

        firstname = request.POST['firstname']
        lastname = request.POST['lastname']
        username = request.POST['username']
        email = request.POST['email']
        password = request.POST['password']
        confirm_password = request.POST['confirm_password']

        if password == confirm_password:
            if User.objects.filter(username=username).exists():
                messages.error(request, 'Username already exists.')
                return redirect('register')
            else:
                if User.objects.filter(email=email).exists():
                    messages.error(request, 'Email already exists.')
                    return redirect('register')
                else:
                    # if all conditionss are satisfied, we save the new user
                    user = User.objects.create_user(
                        first_name=firstname, last_name=lastname, username=username, email=email, password=password)
                    auth.login(request, user)
                    # if you want the user to be logged in automatically
                    # messages.success(request, 'You are now logged in.')
                    # return redirect('dashboard')
                    user.save()
                    messages.success(
                        request, 'You are successfully registered.')
                    return redirect('login')
        else:
            messages.error(request, 'Passwords dont match.')
            return redirect('register')

        # # this calls the error message functionality with INCLUDES\MESSAGES.HTML
        # messages.error(request, 'This is error message')
        return redirect('register')
    else:
        return render(request, 'accounts/register.html')
```

2. Update your REGISTER.HTML



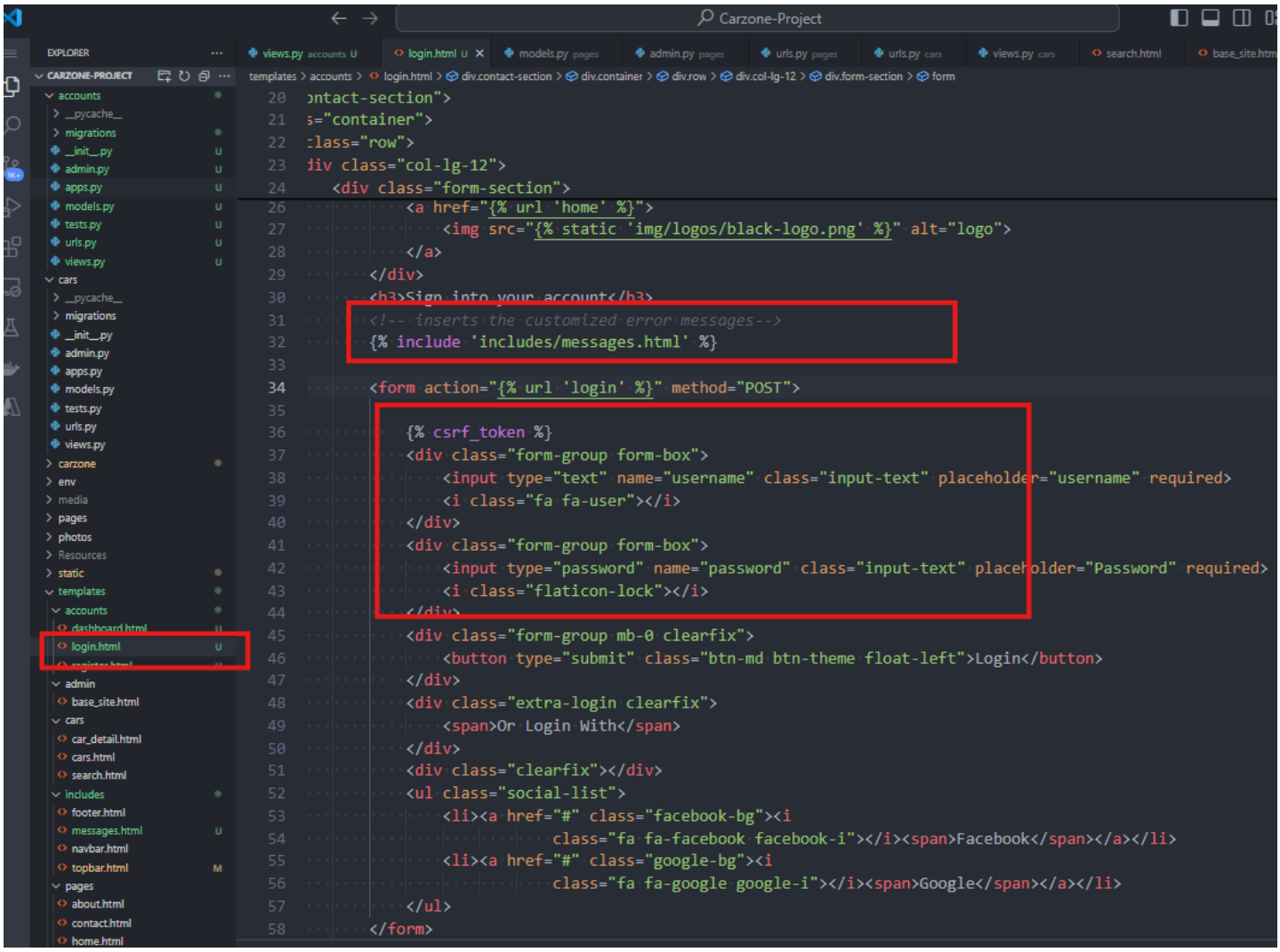
3. You can test it. We updated our ViEWS.PY if we want the user to be logged in automatically or not.

4. Update the LOGIN.HTML as

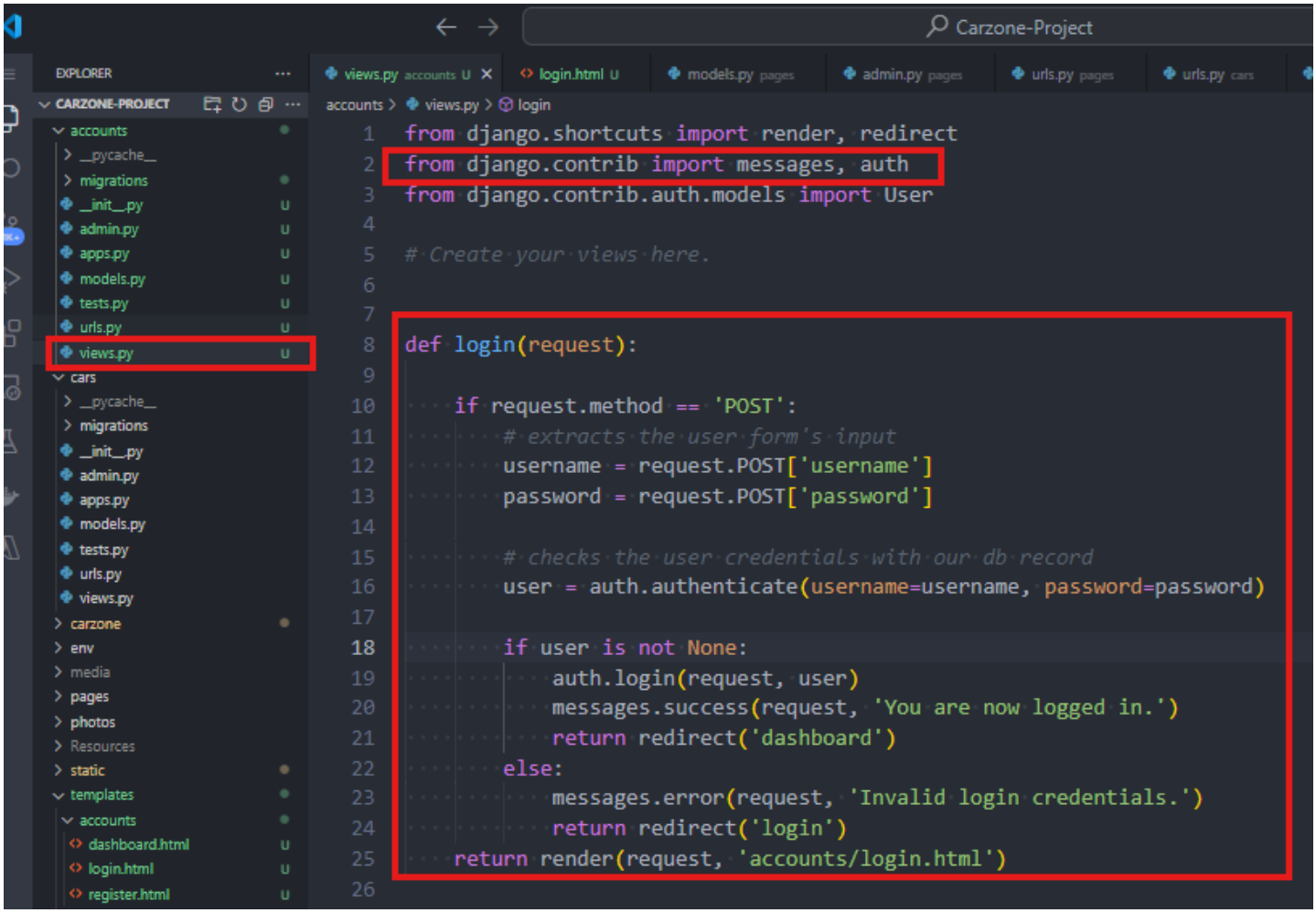
FROM:

```
<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>
```

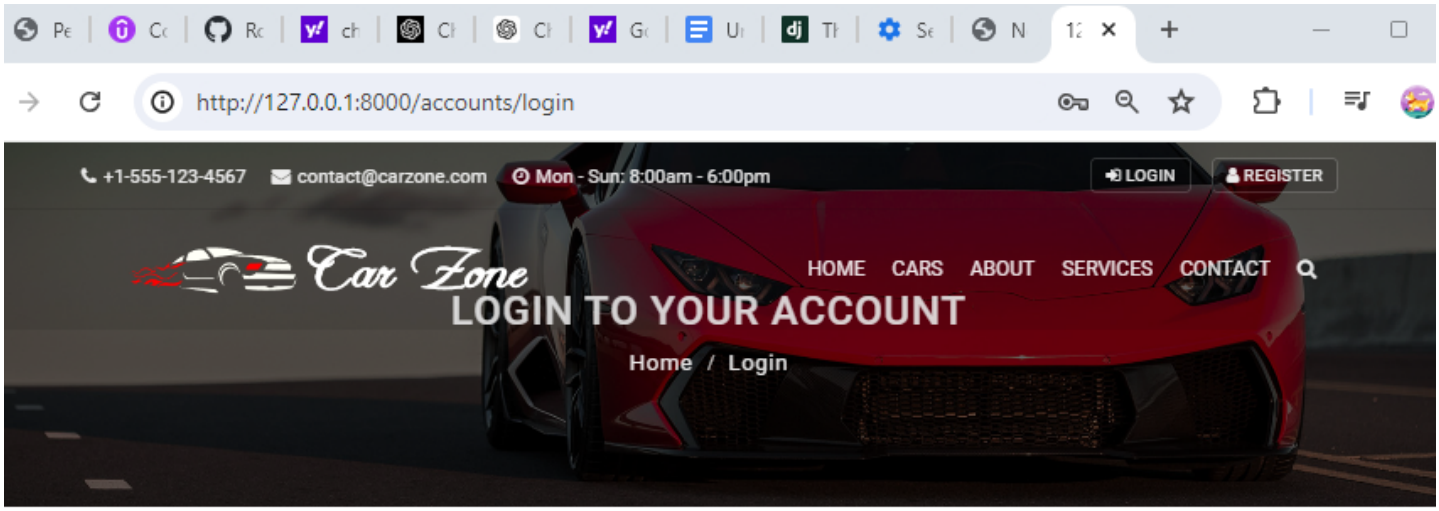
TO:

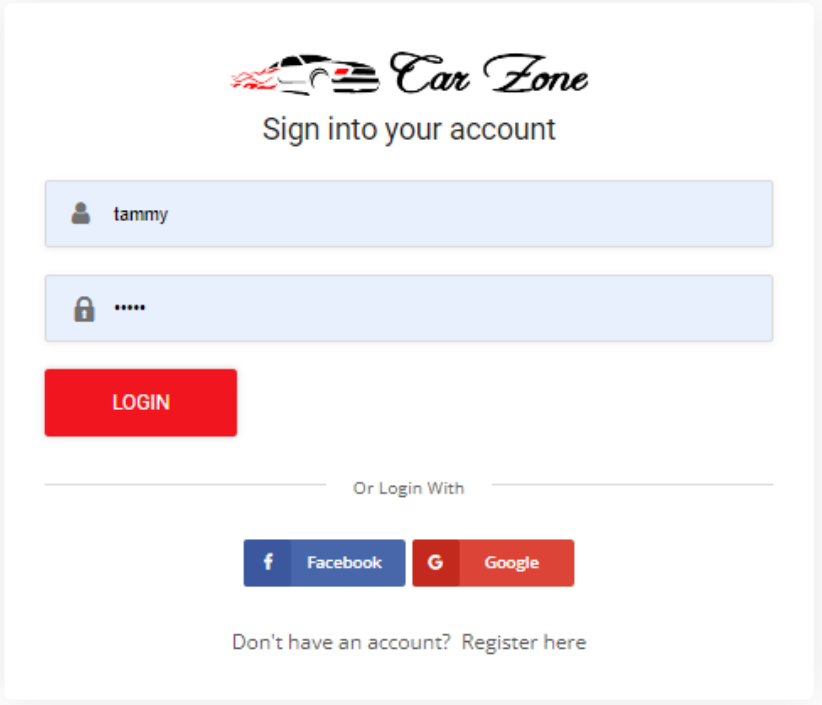


5. Now, for the LOGIN function, update our VIEWS.PY as



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.

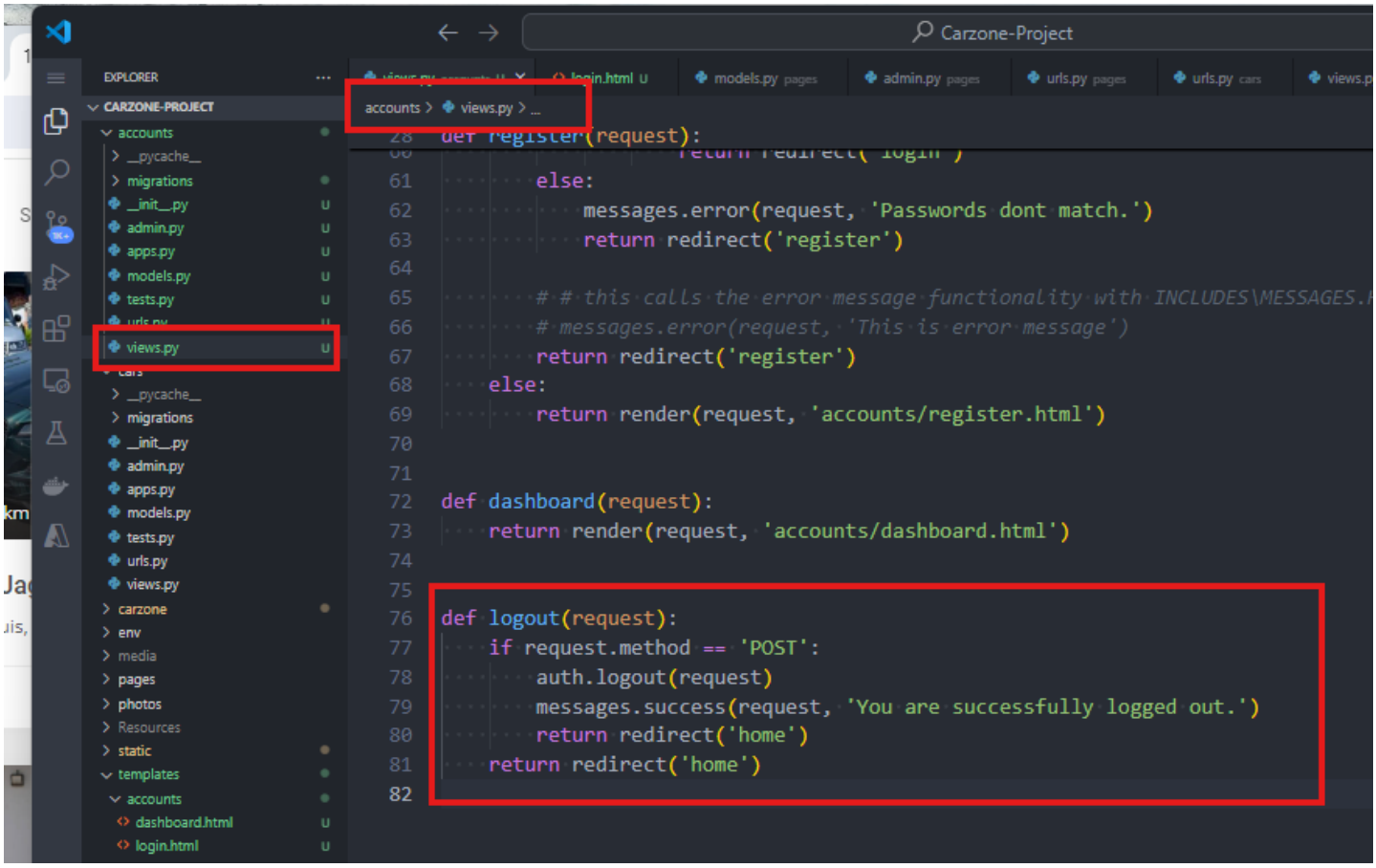
FROM:

```
<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>
```

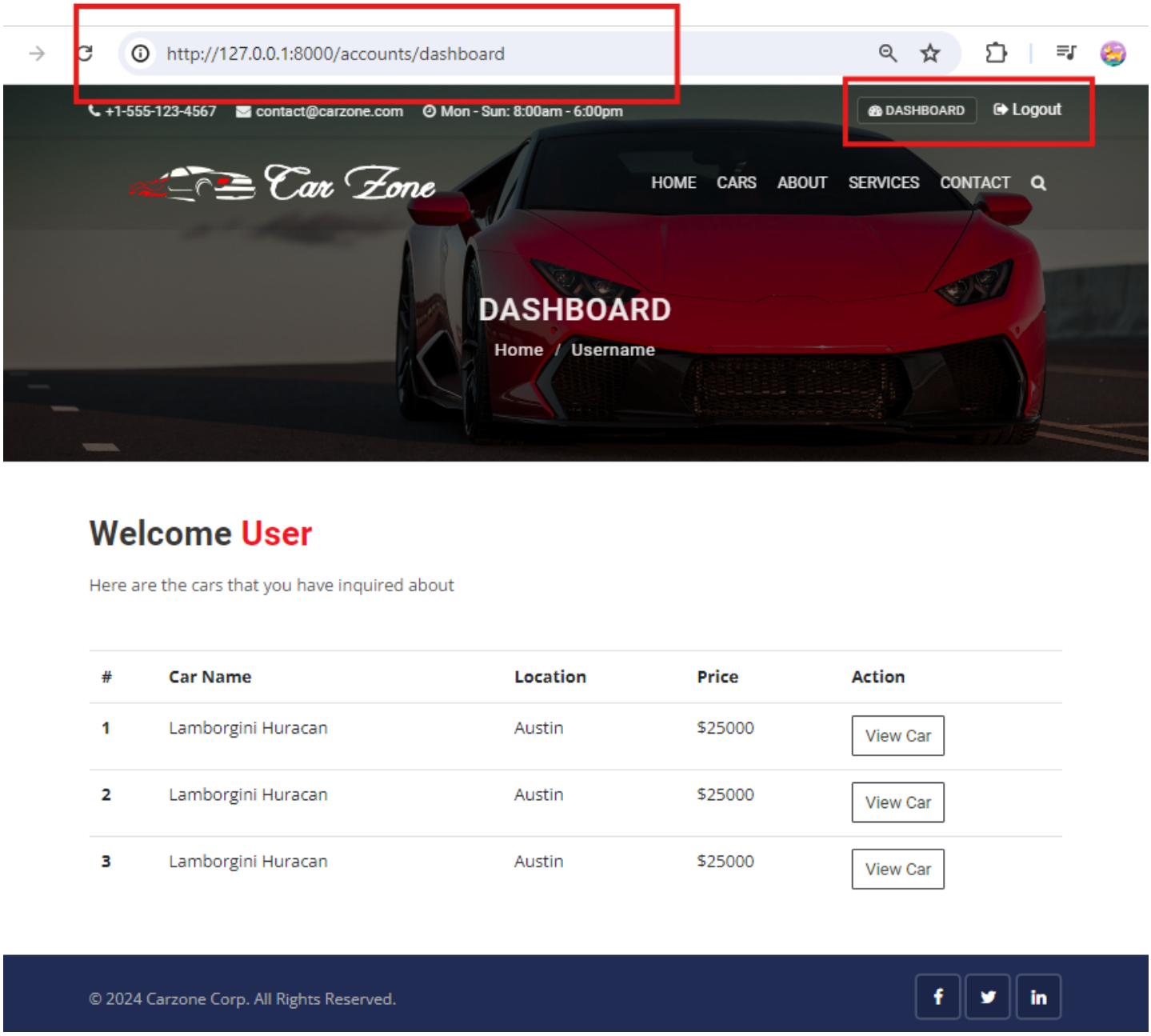
TO:

```
<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 %}
```

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



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



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:

TO:

11. Then in each webpage like PAGES\HOME.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:

TO:

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

Copyright © Personal Digital Notebooks | By Rosilie | Date Printed: Feb. 4, 2026, 2:56 p.m.