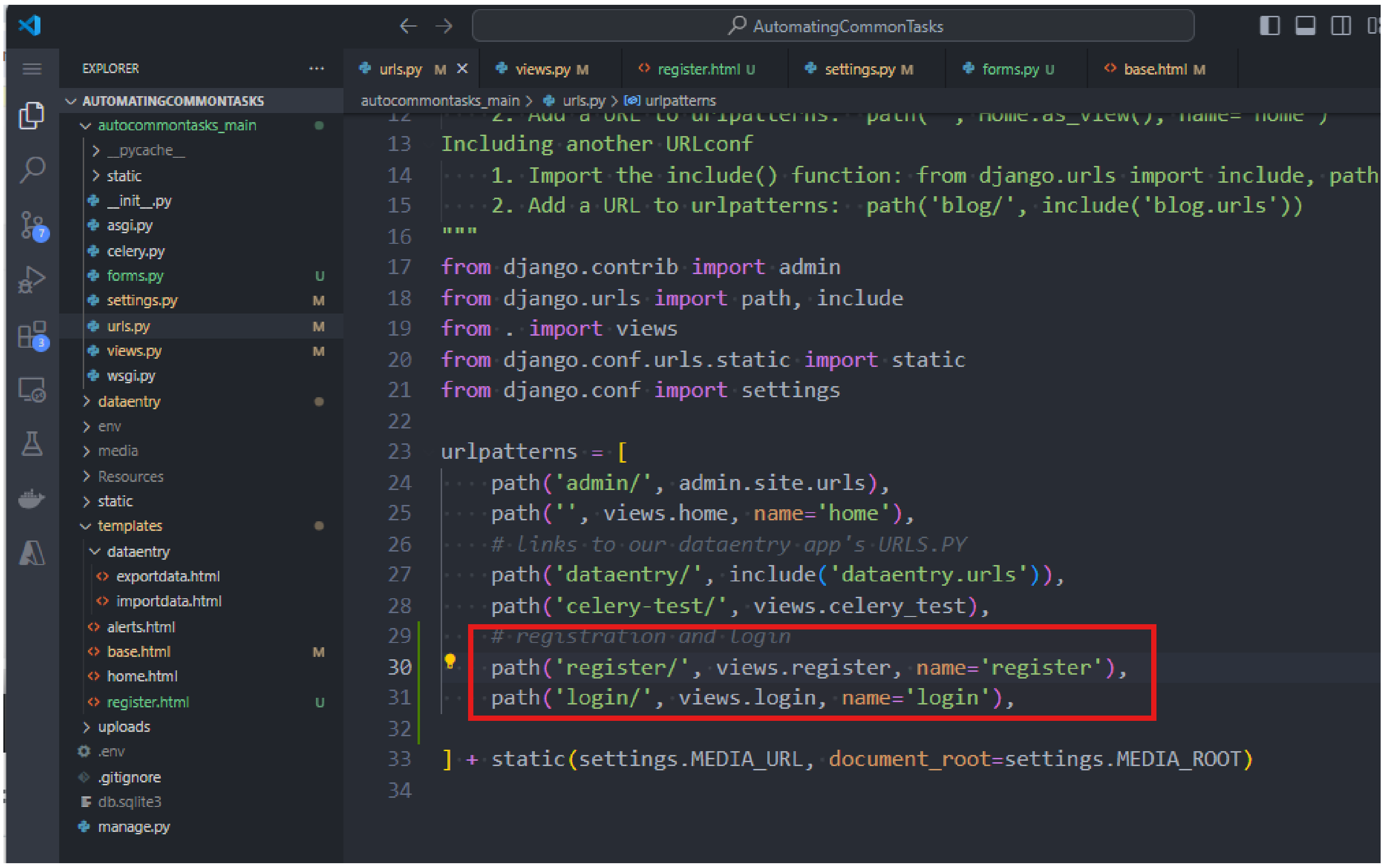
Topic: Login / Logout & User Restrictions Module Part 15

Speaker: Udemy Instructor Rathan Kumar | Notebook: Django: Automating Common Tasks

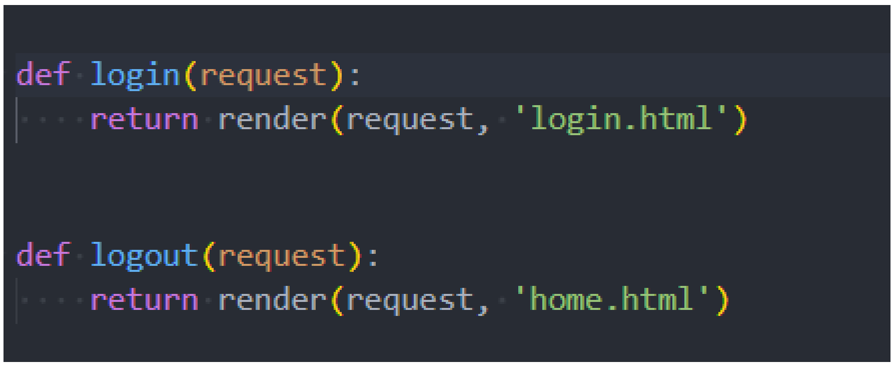
---

1. We completed the REGISTER module previously. To make your Login link work, we need to update our URLS.PY



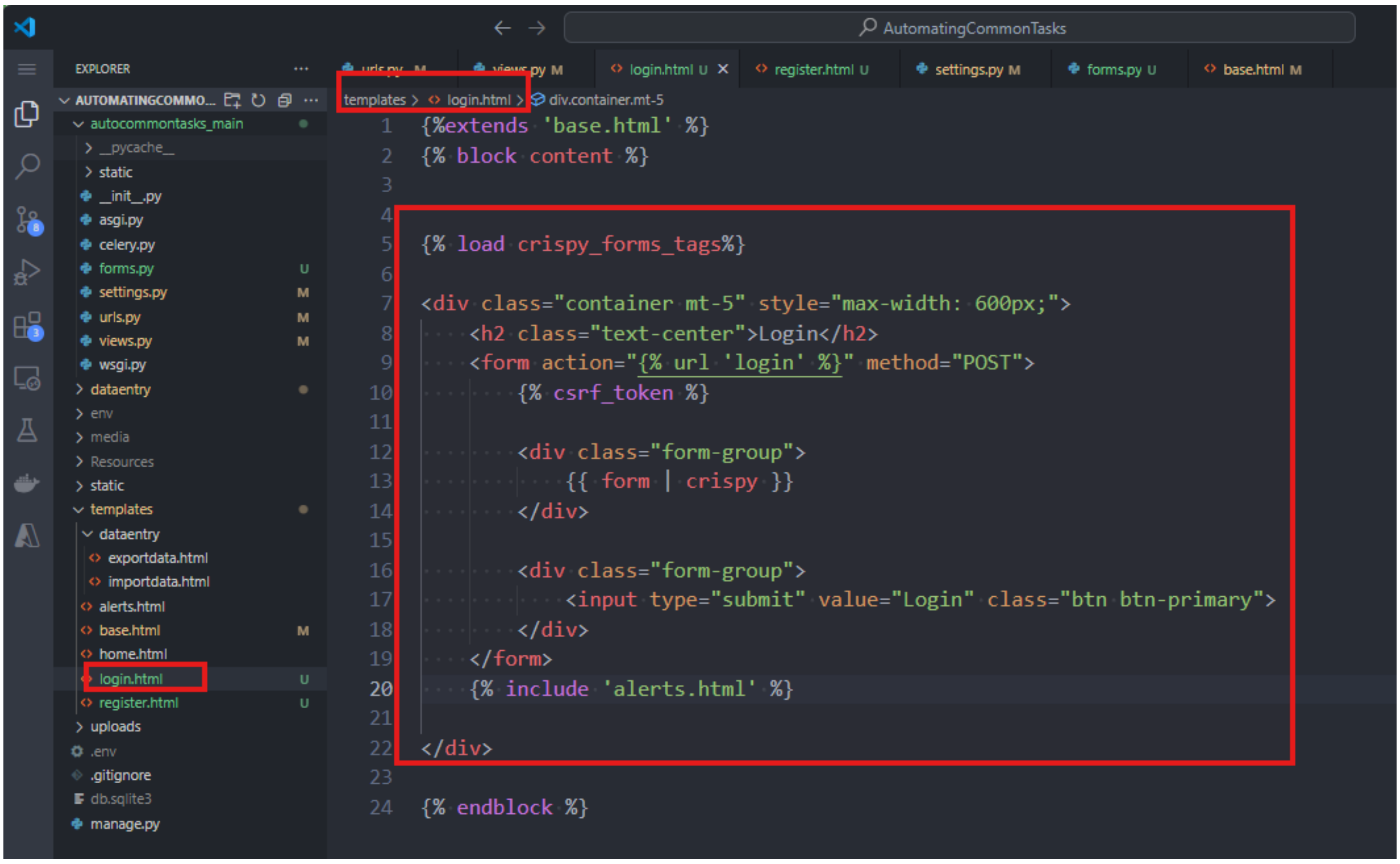
```python
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""

from django.contrib import admin
from django.urls import path, include
from . import views
from django.conf.urls.static import static
from django.conf import settings

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home, name='home'),
    # links to our dataentry app's URLS.PY
    path('dataentry/', include('dataentry.urls')),
    path('celery-test/', views.celery_test),
    # registration and login
    path('register/', views.register, name='register'),
    path('login/', views.login, name='login'),

] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

2. Create our LOGIN function in the VIEWS.PY



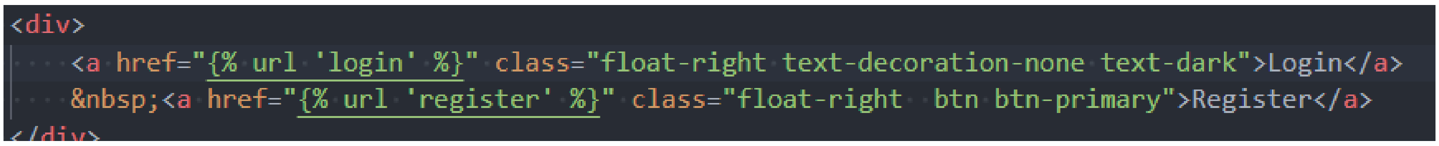
```python
def login(request):
    return render(request, 'login.html')


def logout(request):
    return render(request, 'home.html')
```

3. Create our LOGIN.HTML



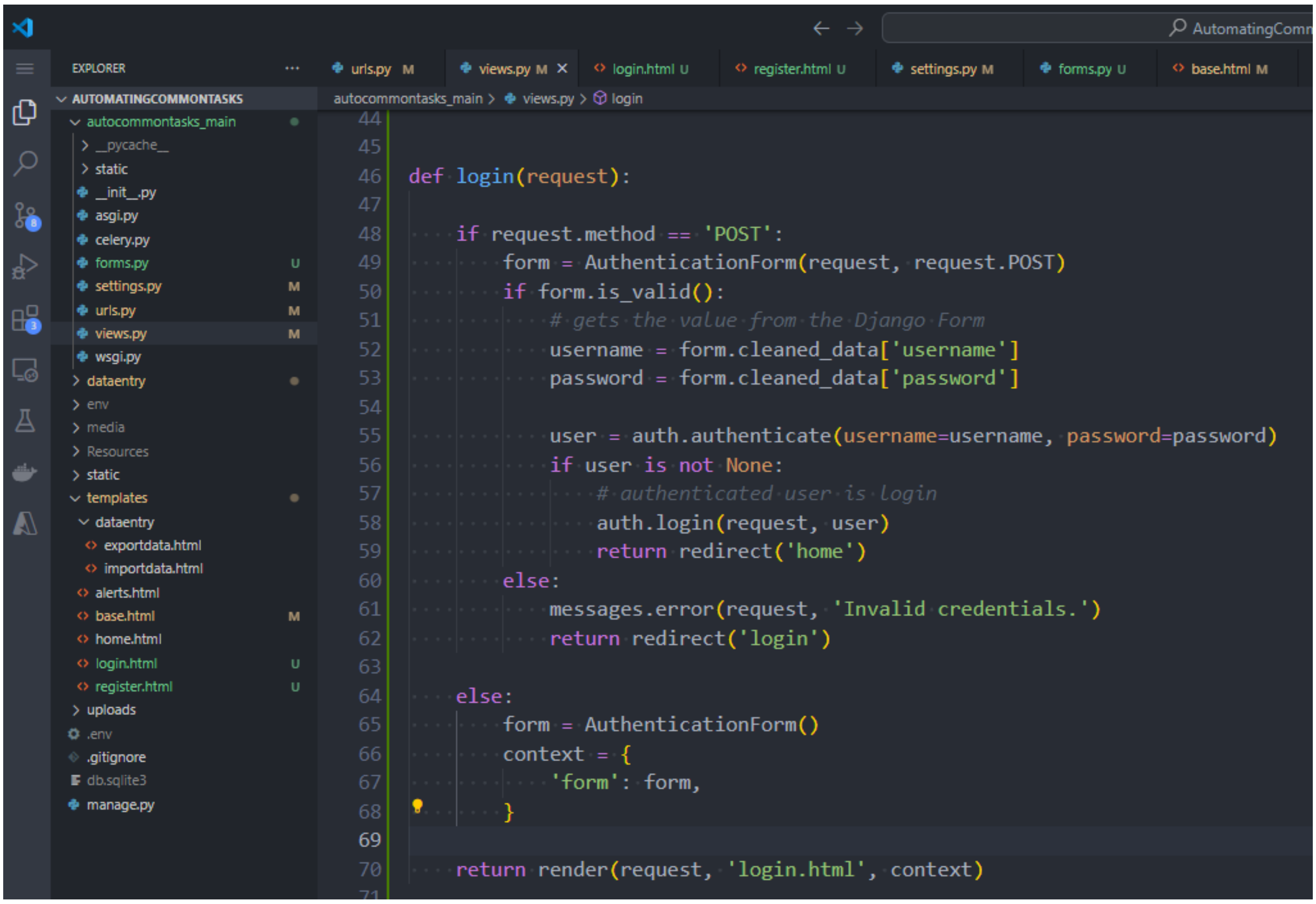
```
{%extends 'base.html' %}
{% block content %}


{% load crispy_forms_tags%}

<div class="container mt-5" style="max-width: 600px;">
    <h2 class="text-center">Login</h2>
    <form action="{% url 'login' %}" method="POST">
        {% csrf_token %}

        <div class="form-group">
            {{ form | crispy }}
        </div>

        <div class="form-group">
            <input type="submit" value="Login" class="btn btn-primary">
        </div>
    </form>
    {% include 'alerts.html' %}

</div>

{% endblock %}
```

4. Update our BASE.HTML



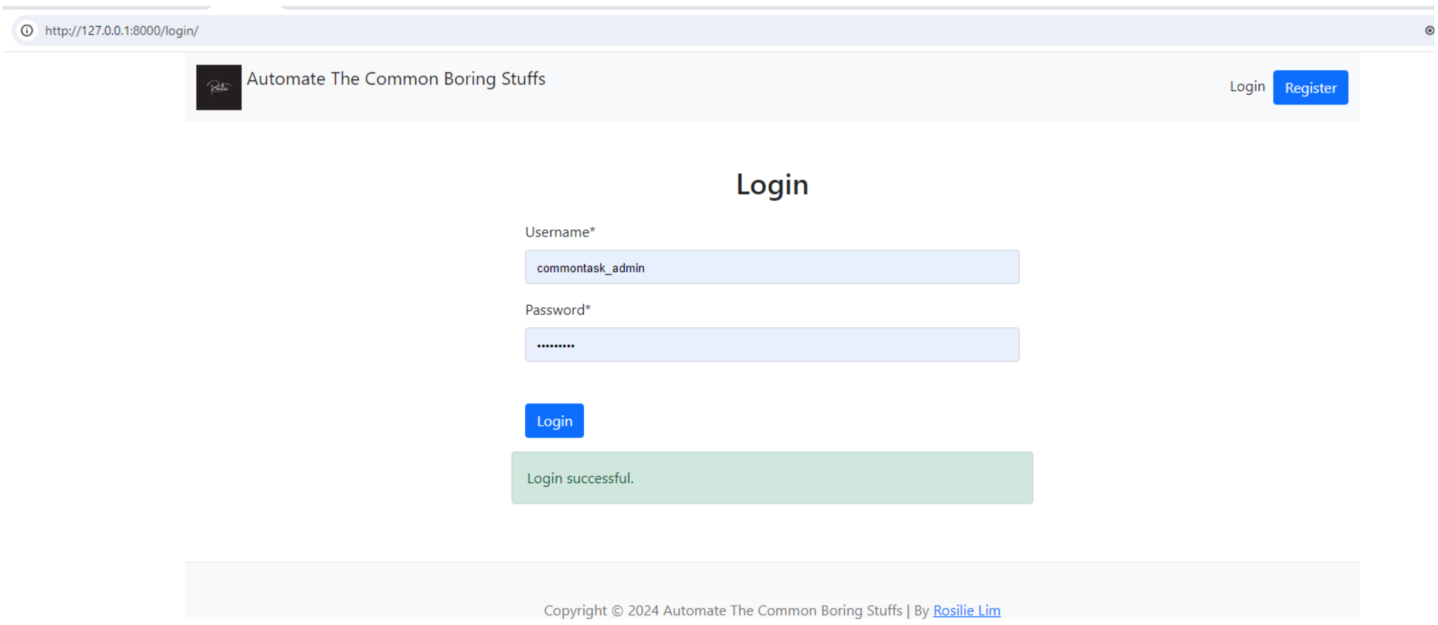
```
<div>
    <a href="{% url 'login' %}" class="float-right text-decoration-none text-dark">Login</a>
    &nbsp;<a href="{% url 'register' %}" class="float-right  btn btn-primary">Register</a>
</div>
```

5. Update our VIEWS.PY



```python
def login(request):

    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        if form.is_valid():
            # gets the value from the Django Form
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']

            user = auth.authenticate(username=username, password=password)
            if user is not None:
                # authenticated user is login
                auth.login(request, user)
                return redirect('home')
        else:
            messages.error(request, 'Invalid credentials.')
            return redirect('login')

    else:
        form = AuthenticationForm()
        context = {
            'form': form,
        }

    return render(request, 'login.html', context)
```

Run the code:


http://127.0.0.1:8000/login/
Automate The Common Boring Stuffs
Login
Register

Login

Username*

commontask_admin

Password*

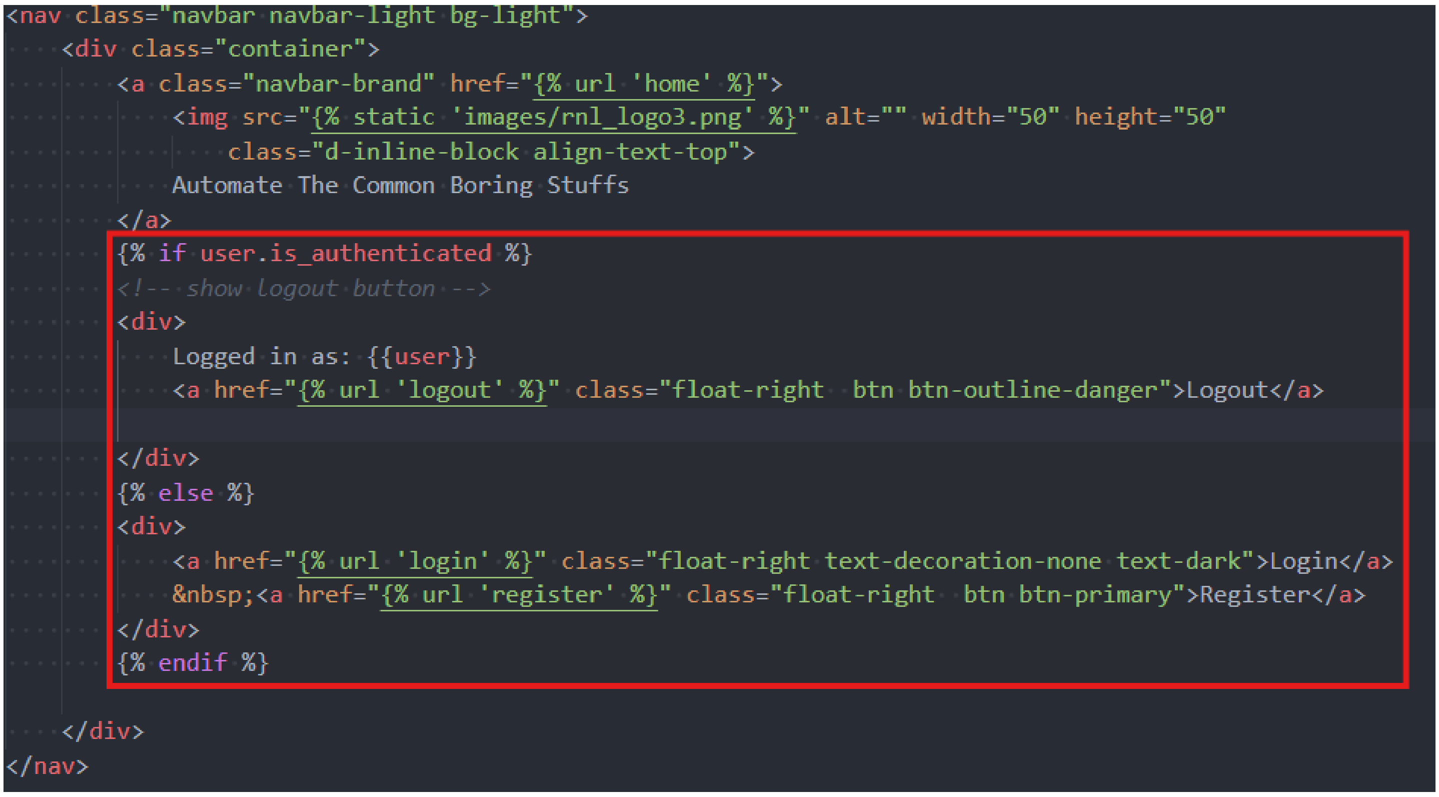
....

Login

Login successful.

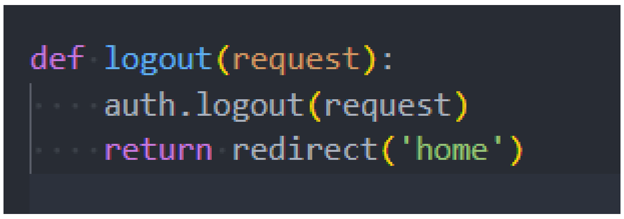
Copyright © 2024 Automate The Common Boring Stuffs | By Rosilie Lim

6. We should not show REGISTER button if we are logged in, so we update our BASE.HTML AS:



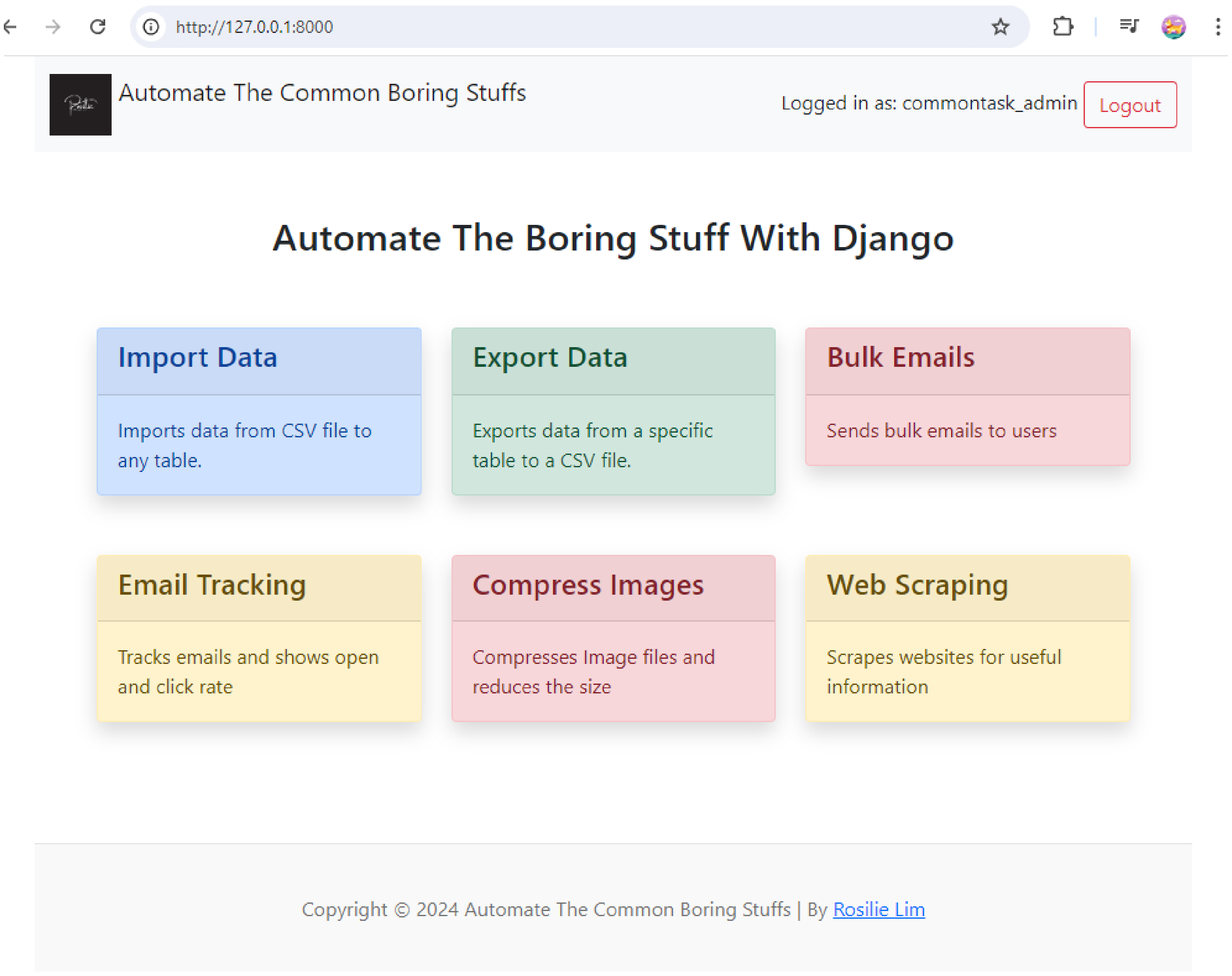
```
<nav class="navbar navbar-light bg-light">
    <div class="container">
        <a class="navbar-brand" href="{% url 'home' %}">
            <img src="{% static 'images/rnl_logo3.png' %}" alt="" width="50" height="50"
                class="d-inline-block align-text-top">
            Automate The Common Boring Stuffs
        </a>
        {% if user.is_authenticated %}
        <!-- show logout button -->
        <div>
            Logged in as: {{user}}
            <a href="{% url 'logout' %}" class="float-right  btn btn-outline-danger">Logout</a>

        </div>
        {% else %}
        <div>
            <a href="{% url 'login' %}" class="float-right text-decoration-none text-dark">Login</a>
            &nbsp;<a href="{% url 'register' %}" class="float-right  btn btn-primary">Register</a>
        </div>
        {% endif %}

    </div>
</nav>
```

7. To make our logout button work, we update our VIEWS.PY AS:



```
def logout(request):
    auth.logout(request)
    return redirect('home')
```

8. Test our code:


http://127.0.0.1:8000

Automate The Common Boring Stuffs

Logged in as: commontask_admin

Logout

Automate The Boring Stuff With Django

Import Data

Imports data from CSV file to any table.

Export Data

Exports data from a specific table to a CSV file.

Bulk Emails

Sends bulk emails to users

Email Tracking

Tracks emails and shows open and click rate

Compress Images

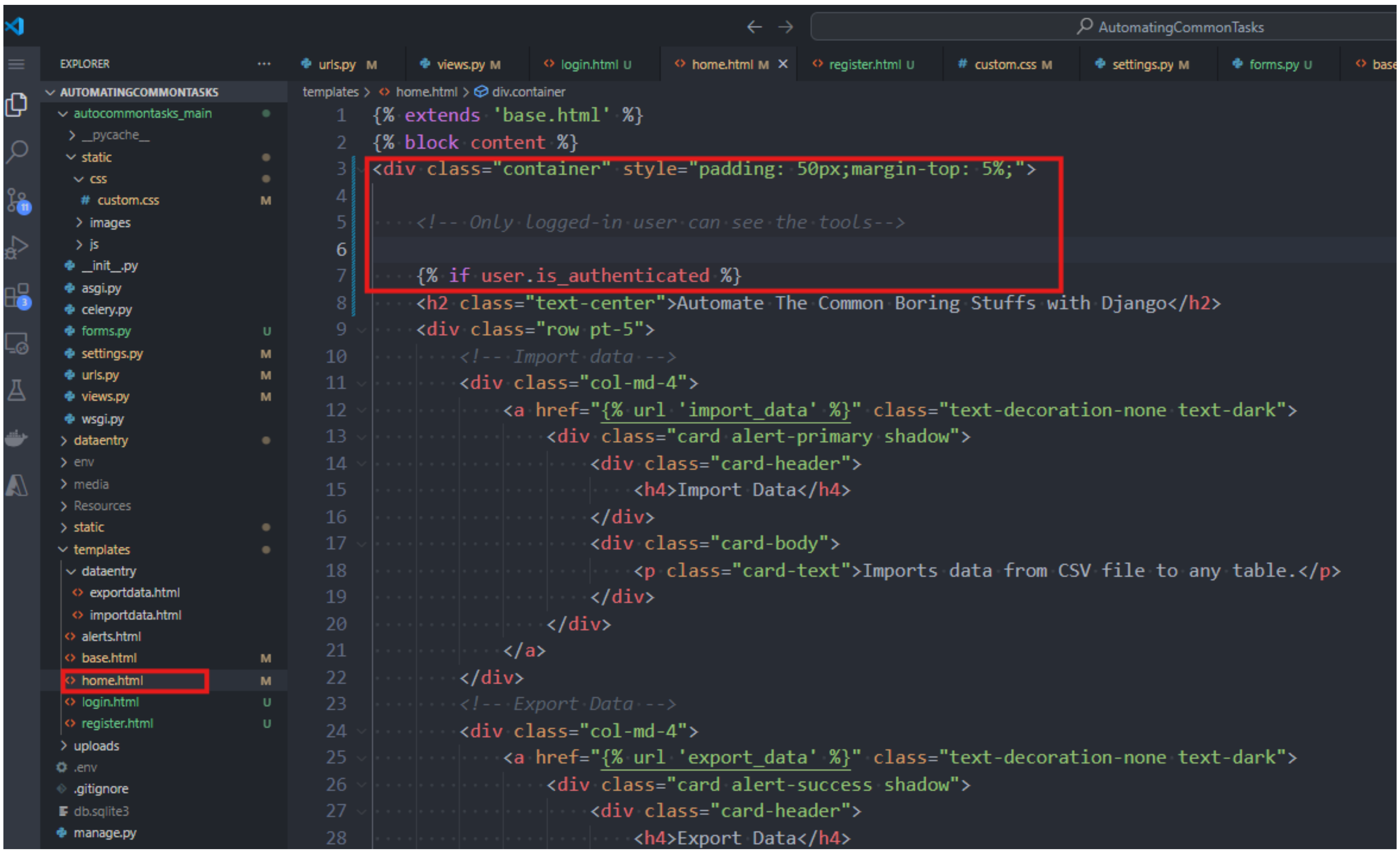
Compresses Image files and reduces the size

Web Scraping

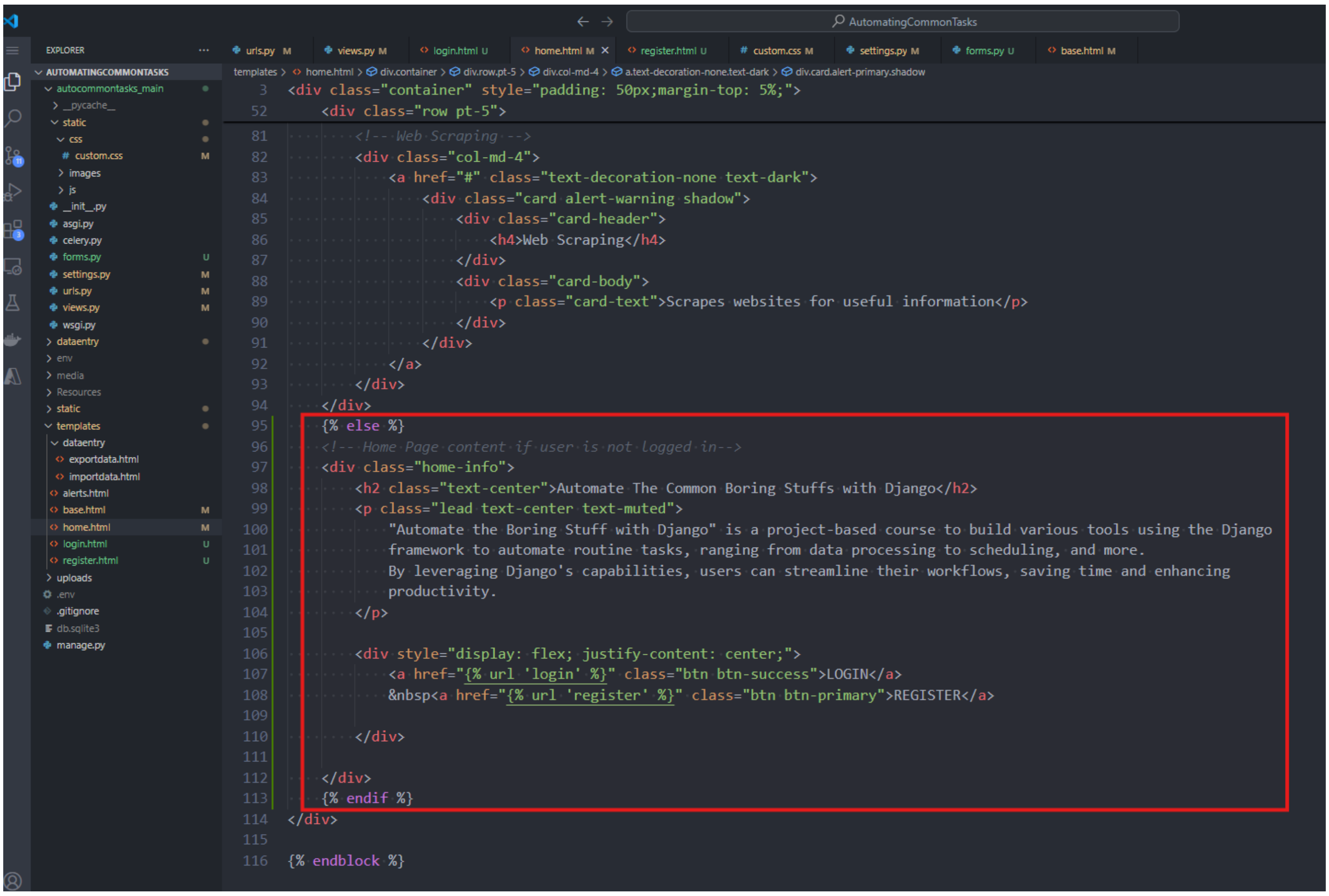
Scrapes websites for useful information

Copyright © 2024 Automate The Common Boring Stuffs | By Rosilie Lim

9. To make see only the tools when they log in, we must update our HOME.HTML. We add the IF ELSE statement so that the tools would only appear if we log in otherwise we see the homepage with some information instead.

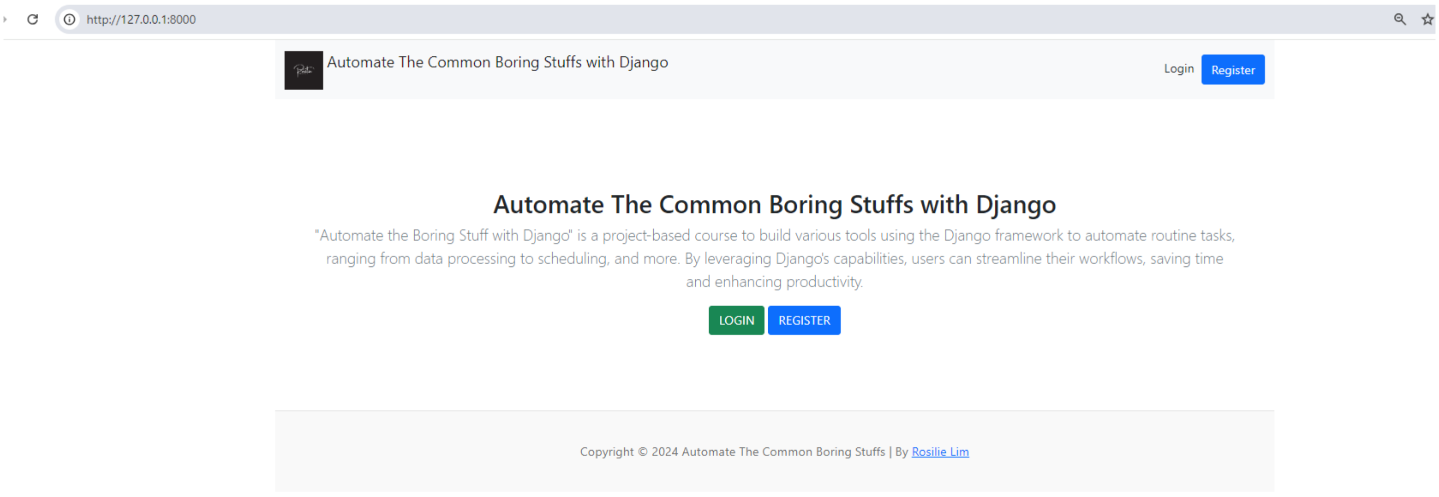


```
{% extends 'base.html' %}
{% block content %}
<div class="container" style="padding: 50px;margin-top: 5%;">

    <!-- Only logged-in user can see the tools-->

    {% if user.is_authenticated %}
    <h2 class="text-center">Automate The Common Boring Stuffs with Django</h2>
    <div class="row pt-5">
        <!-- Import data -->
        <div class="col-md-4">
            <a href="{% url 'import_data' %}" class="text-decoration-none text-dark">
                <div class="card alert-primary shadow">
                    <div class="card-header">
                        <h4>Import Data</h4>
                    </div>
                    <div class="card-body">
                        <p class="card-text">Imports data from CSV file to any table.</p>
                    </div>
                </div>
            </a>
        </div>
        <!-- Export Data -->
        <div class="col-md-4">
            <a href="{% url 'export_data' %}" class="text-decoration-none text-dark">
                <div class="card alert-success shadow">
                    <div class="card-header">
                        <h4>Export Data</h4>
```



```
<div class="container" style="padding: 50px;margin-top: 5%;">
    <div class="row pt-5">
        <!-- Web Scraping -->
        <div class="col-md-4">
            <a href="#" class="text-decoration-none text-dark">
                <div class="card alert-warning shadow">
                    <div class="card-header">
                        <h4>Web Scraping</h4>
                    </div>
                    <div class="card-body">
                        <p class="card-text">Scrapes websites for useful information</p>
                    </div>
                </div>
            </a>
        </div>
    </div>
    {% else %}
    <!-- Home Page content if user is not logged in-->
    <div class="home-info">
        <h2 class="text-center">Automate The Common Boring Stuffs with Django</h2>
        <p class="lead text-center text-muted">
            "Automate the Boring Stuff with Django" is a project-based course to build various tools using the Django
            framework to automate routine tasks, ranging from data processing to scheduling, and more.
            By leveraging Django's capabilities, users can streamline their workflows, saving time and enhancing
            productivity.
        </p>

        <div style="display: flex; justify-content: center;">
            <a href="{% url 'login' %}" class="btn btn-success">LOGIN</a>
            &nbsp<a href="{% url 'register' %}" class="btn btn-primary">REGISTER</a>

        </div>

    </div>
    {% endif %}
</div>

{% endblock %}
```

9. Load the page:


http://127.0.0.1:8000

Automate The Common Boring Stuffs with Django

Login Register

Automate The Common Boring Stuffs with Django

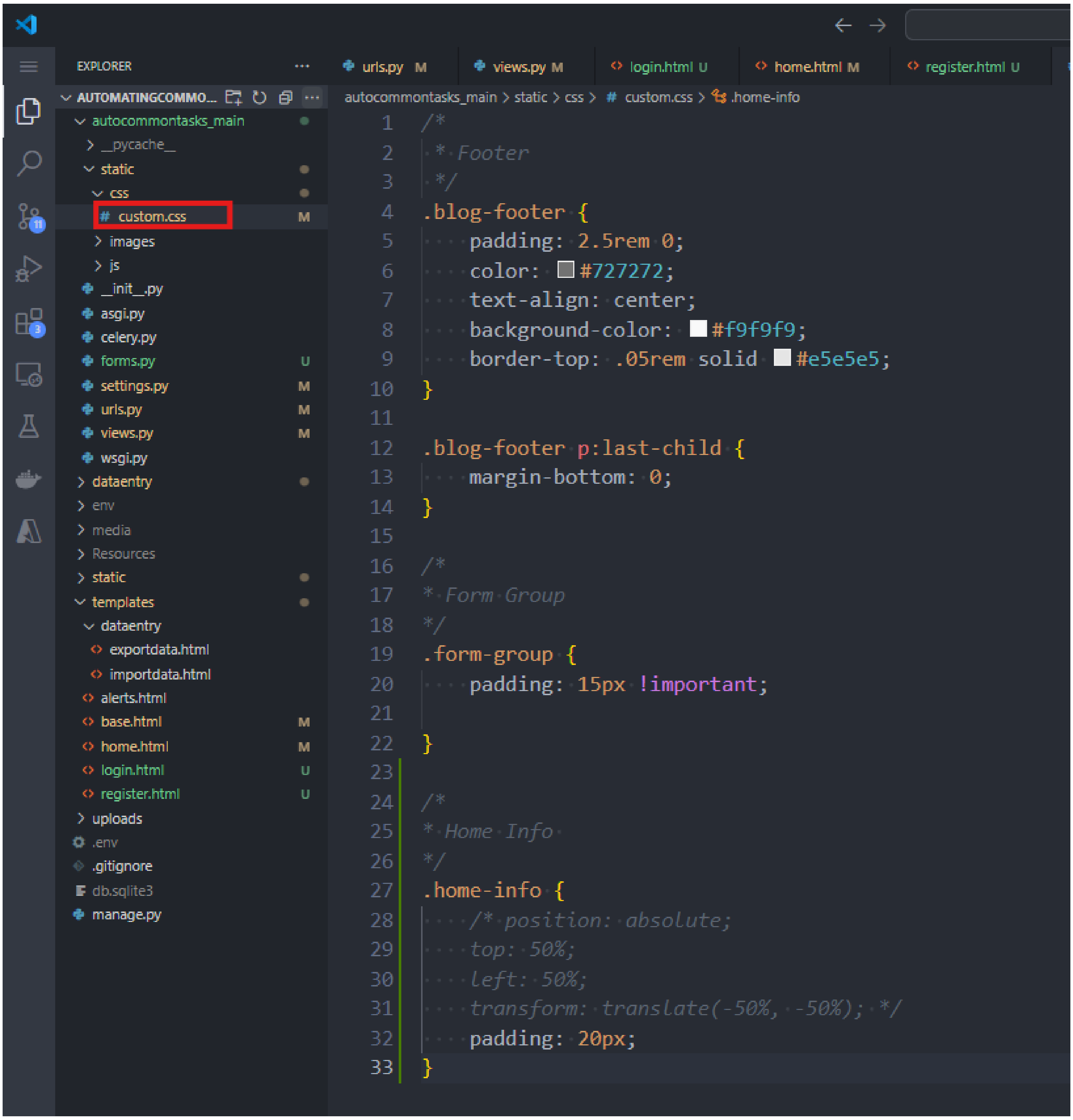
"Automate the Boring Stuff with Django" is a project-based course to build various tools using the Django framework to automate routine tasks, ranging from data processing to scheduling, and more. By leveraging Django's capabilities, users can streamline their workflows, saving time and enhancing productivity.

LOGIN REGISTER

Copyright © 2024 Automate The Common Boring Stuffs | By Rosilie Lim

From here, you can login or register to see the Django tools.

We have made some custom CSS for our footer and home-info: We run COLLECTSTATIC to make sure that this external CSS is added to our root folder for deployment.



```css
/*
 * Footer
 */
.blog-footer {
    padding: 2.5rem 0;
    color: #727272;
    text-align: center;
    background-color: #f9f9f9;
    border-top: .05rem solid #e5e5e5;
}

.blog-footer p:last-child {
    margin-bottom: 0;
}

/*
* Form Group
*/
.form-group {
    padding: 15px !important;

}

/*
* Home Info
*/
.home-info {
    /* position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%); */
    padding: 20px;
}
```