

Django practice setup for Signup, Login, Logout using `ModelForm`. This is a simple, beginner-friendly version you can plug into a project.

□ 1. Create Project + App

```
django-admin startproject myproject
cd myproject
python manage.py startapp accounts
```

Add app in `settings.py`:

```
INSTALLED_APPS = [
    ...
    'accounts',
]
```

□ 2. Use Django User Model

We'll use built-in:

```
from django.contrib.auth.models import User
```

□ 3. forms.py (ModelForms)

□ `accounts/forms.py`

```
from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import AuthenticationForm

# SIGNUP FORM (ModelForm)
class SignUpForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)
    confirm_password = forms.CharField(widget=forms.PasswordInput)

    class Meta:
        model = User
        fields = ['username', 'email', 'password']

    def clean(self):
        cleaned_data = super().clean()
        password = cleaned_data.get("password")
        confirm_password = cleaned_data.get("confirm_password")
```

```
if password != confirm_password:
    raise forms.ValidationError("Passwords do not match")
```

□ 4. views.py

□ accounts/views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.forms import AuthenticationForm
from .forms import SignUpForm

# SIGNUP
def signup_view(request):
    if request.method == "POST":
        form = SignUpForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.set_password(form.cleaned_data['password']) # hash password
            user.save()
            return redirect('login')
    else:
        form = SignUpForm()

    return render(request, 'signup.html', {'form': form})

# LOGIN
def login_view(request):
    if request.method == "POST":
        form = AuthenticationForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            return redirect('home')
    else:
        form = AuthenticationForm()

    return render(request, 'login.html', {'form': form})

# LOGOUT
def logout_view(request):
    logout(request)
    return redirect('login')

# HOME (after login)
def home(request):
    return render(request, 'home.html')
```

□ 5. urls.py

□ accounts/urls.py

```
from django.urls import path
from .views import signup_view, login_view, logout_view, home

urlpatterns = [
    path('', home, name='home'),
    path('signup/', signup_view, name='signup'),
    path('login/', login_view, name='login'),
    path('logout/', logout_view, name='logout'),
]
```

□ main urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('accounts.urls')),
]
```

□ 6. Templates

Create folder:

accounts/templates/

signup.html

```
<h2>Signup</h2>

<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Signup</button>
</form>

<a href="{% url 'login' %}">Login</a>
```

login.html

```
<h2>Login</h2>

<form method="POST">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Login</button>
</form>

<a href="{% url 'signup' %}">Signup</a>
```

home.html

```
{% if user.is_authenticated %}

    <h1> Welcome {{ user.username }}</h1>

    <a href="{% url 'logout' %}">Logout</a>

{% else %}

    <a href="{% url 'login' %}">Login</a>

{% endif %}
```

□ 7. Run migrations + server

```
python manage.py migrate
python manage.py runserver
```

□ What you learned here

- ModelForm for signup
- Password hashing with `set_password()`
- Built-in `AuthenticationForm`
- Django auth system (`login`, `logout`)
- Session-based login

Signup :-

ChatGPT x 127.0.0.1:8000/signup/ x + Ask Gemini - □ ×

← → 127.0.0.1:8000/signup/ ☆ 📄 📄 📄 | 🔒 Verify that it's you Action required ⋮

Signup

Username: Required: 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Email address:

Password:

Confirm password:

[Login](#)

Activate Windows
Go to Settings to activate Windows.

Windows search bar: Type here to search

Taskbar icons: File Explorer, Edge, Chrome, VS Code, etc.

System tray: 34°C Sunny, 4:30 PM 5/9/2026

Login:-

ChatGPT x 127.0.0.1:8000/login/ x + Ask Gemini - □ ×

← → 127.0.0.1:8000/login/ ☆ 📄 📄 📄 | 🔒 Verify that it's you Action required ⋮

Login

Username:

Password:

[Signup](#)

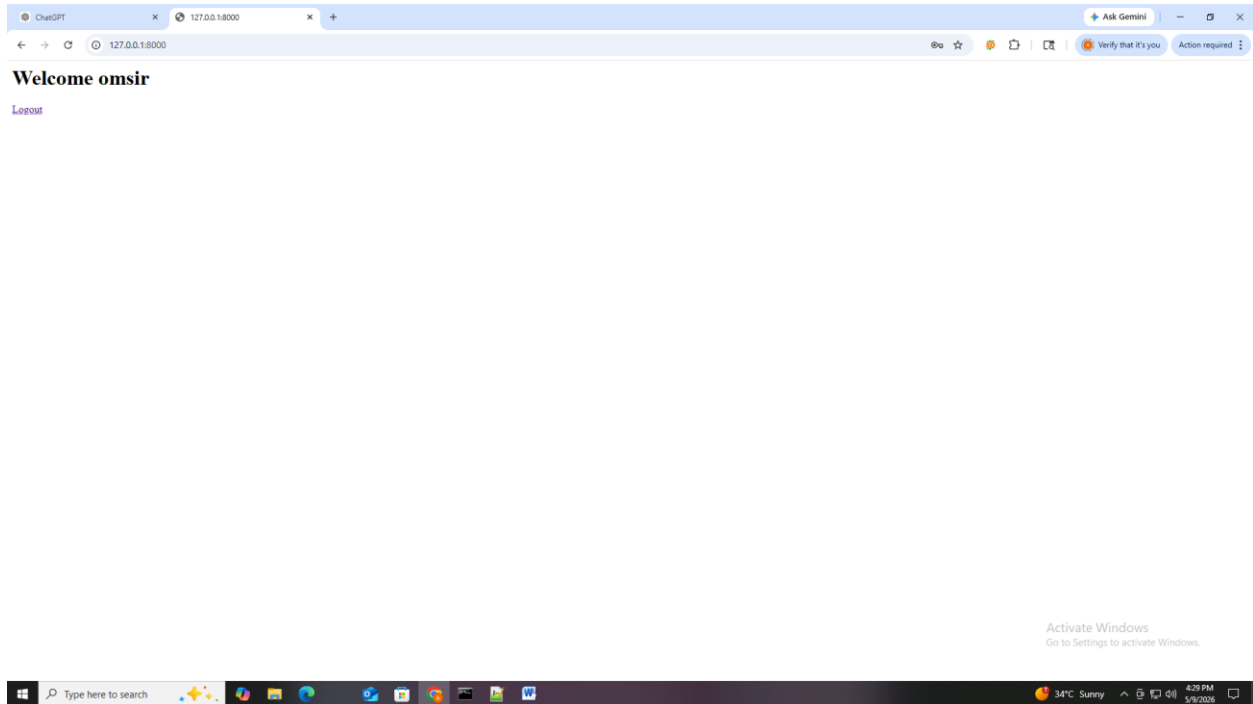
Activate Windows
Go to Settings to activate Windows.

Windows search bar: Type here to search

Taskbar icons: File Explorer, Edge, Chrome, VS Code, etc.

System tray: 34°C Sunny, 4:30 PM 5/9/2026

And After Login



Code meaning :-

```
# SIGNUP
def signup_view(request):
```

➡ Creates a function (view) for signup page.
request contains user request data.

```
if request.method == "POST":
```

➡ Checks if form was submitted.
POST = user clicked submit button.

```
form = SignUpForm(request.POST)
```

➡ Takes submitted form data and puts it into the form.

```
if form.is_valid():
```

➡ Checks:

- all fields filled correctly
- no validation errors

```
user = form.save(commit=False)
```

➡ Creates user object but does NOT save to database yet.
We pause saving because we want to hash password first.

```
user.set_password(form.cleaned_data['password'])
```

➡ Converts normal password into hashed password.

```
user.save()
```

➡ Saves user into database.

```
return redirect('login')
```

➡ After signup, sends user to login page.

```
else:  
    form = SignUpForm()
```

➡ If page is opened normally (GET request), show empty signup form.

```
return render(request, 'signup.html', {'form': form})
```

➡ Sends form to HTML page.

- `signup.html` = template file
 - `{'form': form}` = sends form variable to template
-

LOGIN VIEW

```
def login_view(request):
```

➡ Creates login page function.

```
if request.method == "POST":
```

➡ Checks if login form submitted.

```
form = AuthenticationForm(data=request.POST)
```

➡ Django built-in login form.
Checks username + password.

```
if form.is_valid():
```

➡ Checks if login details are correct.

```
user = form.get_user()
```

➡ Gets logged-in user object.

```
login(request, user)
```

➡ Logs user into session.
Now Django remembers user.

```
return redirect('home')
```

➡ Sends user to home page after login.

```
else:
```

```
    form = AuthenticationForm()
```

➡ If page opened normally, show empty login form.

```
return render(request, 'login.html', {'form': form})
```


➡ Sends login form to HTML page.

LOGOUT VIEW

```
def logout_view(request):
```

➡ Creates logout function.

```
logout(request)
```

➡ Removes user session.
User becomes logged out.

```
return redirect('login')
```

➡ After logout, sends user to login page.

HOME VIEW

```
def home(request):
```

➡ Creates home page function.

```
return render(request, 'home.html')
```

➡ Opens `home.html` page.