**Video file : 101**

**Code file : ch91**

**Builtin Authentication in Django 5**

### 🧠 **1. Introduction**

Django provides a **built-in authentication system** that allows us to easily implement user login, logout, password change, and password reset features without creating everything from scratch.

### 🏗 **2. Basic Setup Using Built-in Templates**

#### Step 1: Create a New Django Project

Create your project and then open the main urls.py file. Add the following code to include Django’s authentication URLs:

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

    path('admin/', admin.site.urls),

    path('accounts/', include('django.contrib.auth.urls')),

]

#### Step 2: Apply Migrations

Run the following commands to apply the required authentication migrations:

python manage.py makemigrations

python manage.py migrate

#### Step 3: Run the Development Server

python manage.py runserver

#### Step 4: Access the Login Page

Open your browser and go to:

http://127.0.0.1:8000/accounts/login/

Django will look for the template:

registration/login.html

So, we need to create an app named **registration** (or any other name) and place the required templates inside it.

### 📝 **3. Required Templates**

* registration/login.html → Login form
* registration/profile.html → Shown after login

By using these built-in views, Django automatically gives us login, logout, password reset, and password change functionality.

### 🎨 **4. Using Custom Templates (Own App)**

If we want to customize the templates (e.g., design our own login page), we can create our own app.  
Let’s call it **myapp**.

#### ✅ Project urls.py:

from django.contrib import admin

from django.urls import path

from django.contrib.auth import views as auth\_views

urlpatterns = [

    path('admin/', admin.site.urls),

    path('login/', auth\_views.LoginView.as\_view(template\_name='myapp/login.html')),

]

### 🧭 **5. Using a Class-Based View for Custom Login (Optional)**

In myapp/views.py:

from django.contrib.auth.views import LoginView

class CustomLoginView(LoginView):

    template\_name = 'myapp/login.html'

### 🚀 **6. Redirecting After Login**

By default, after login, Django redirects users to /accounts/profile/.  
We can change this by adding the following line in settings.py:

LOGIN\_REDIRECT\_URL = '/dashboard/'

### 🧰 **7. Complete Example**

#### 📄 **Project** urls.py

from django.contrib import admin

from django.urls import path

from django.contrib.auth import views as auth\_views

from django.views.generic import TemplateView

urlpatterns = [

    path('admin/', admin.site.urls),

    path('login/', auth\_views.LoginView.as\_view(template\_name='myapp/login.html'), name='login'),

    path('dashboard/', TemplateView.as\_view(template\_name='myapp/dashboard.html'), name='dashboard'),

    path('logout/', auth\_views.LogoutView.as\_view(template\_name='myapp/logout.html'), name='logout'),

    path('changepass/', auth\_views.PasswordChangeView.as\_view(

        template\_name='myapp/changepass.html',

        success\_url='/changepassdone/'

    ), name='changepass'),

    path('changepassdone/', auth\_views.PasswordChangeDoneView.as\_view(

        template\_name='myapp/changepassdone.html'

    ), name='changepassdone'),

]

### 🧾 **8. Template Files**

#### 🟡 myapp/templates/myapp/login.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Login Page</title>

</head>

<body>

  <h1>Login</h1>

  <form action="" method="post" novalidate>

    {% csrf\_token %}

    {% for fm in form %}

      <div>

        {{ fm.label\_tag }} {{ fm }}

        <small>{{ fm.errors|striptags }}</small>

      </div>

    {% endfor %}

    <input type="submit" value="Login">

    {% if form.non\_field\_errors %}

      {% for error in form.non\_field\_errors %}

        <p>{{ error }}</p>

      {% endfor %}

    {% endif %}

  </form>

</body>

</html>

#### 🟢 myapp/templates/myapp/dashboard.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Dashboard</title>

</head>

<body>

  <h1>Dashboard Page</h1>

  <form action="{% url 'logout' %}" method="post">

    {% csrf\_token %}

    <input type="submit" value="Logout">

  </form>

  <a href="{% url 'changepass' %}">Change Password</a>

</body>

</html>

#### 🔵 myapp/templates/myapp/logout.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Logout</title>

</head>

<body>

  <h1>You have logged out successfully!</h1>

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

</body>

</html>

#### 🟠 myapp/templates/myapp/changepass.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Change Password</title>

</head>

<body>

  <h1>Change Password</h1>

  <form action="" method="post">

    {% csrf\_token %}

    {{ form.as\_p }}

    <input type="submit" value="Save">

    <a href="{% url 'dashboard' %}">Cancel</a>

  </form>

</body>

</html>

#### 🟣 myapp/templates/myapp/changepassdone.html

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <title>Password Changed</title>

</head>

<body>

  <h1>Password Changed Successfully</h1>

  <a href="{% url 'dashboard' %}">Back to Dashboard</a>

</body>

</html>

### 📝 **9. Summary**

* ✅ **django.contrib.auth.urls** gives us ready-made authentication URLs (login, logout, password management).
* 🧭 We can change the **redirect path after login** using LOGIN\_REDIRECT\_URL.
* 🎨 We can customize templates by placing them inside our own app’s templates folder.
* 👤 Password change, password change done, and logout views can also use custom templates.