**Django Authentication: Login, Signup & Logout**

Django provides a built-in authentication system that handles user creation, login, and logout securely and efficiently.

**1. Prerequisites**

* Ensure django.contrib.auth, django.contrib.contenttypes, and django.contrib.sessions are included in INSTALLED\_APPS.
* Templates should be configured with context processors, especially django.contrib.auth.context\_processors.auth.

**2. User Signup (Registration)**

* Use Django's UserCreationForm to register users.
* Display a form where users can enter a username and password.
* On successful submission, the user is created and redirected to a login page or another view.
* You can customize the form or create a new one for additional fields like email or full name.

**3. User Login**

* Authenticate users using Django’s authentication system.
* Use Django's LoginView for a built-in login page or create a custom view.
* On successful login, the user session is created and the user is redirected to a protected page (like dashboard or home).
* Optionally show error messages for incorrect login attempts.

**4. User Logout**

* Use Django’s logout() function or LogoutView to log out a user.
* After logging out, redirect the user to a public page such as the login or home page.

**5. URLs Configuration**

* Map signup, login, and logout views in your urls.py.
* Use Django’s built-in views for login and logout if preferred.
* Assign appropriate name values to the routes for easy referencing in templates and redirects.

**🔏 6. Login Required Access Control**

* Use the @login\_required decorator to protect views.
* Unauthorized users are redirected to the login page.
* Configure LOGIN\_URL, LOGIN\_REDIRECT\_URL, and LOGOUT\_REDIRECT\_URL in settings.py for better control of the flow.