

Tops Technology

Module 16)

Python DB and Framework

Presented By : Nandni Vala

Social Authentication

1. Setting up social login options (Google, Facebook, GitHub) in Django using OAuth2.

- Implementing social login options allows users to log in to your Django application using their Google, Facebook, or GitHub accounts. This can be achieved using Django libraries like **django-allauth**, which simplifies OAuth2 integration.
 - **Install Required Libraries**
 - Install django-allauth for handling social authentication.
 - `pip install django-allauth`
 - **Configure INSTALLED_APPS in settings.py**
 - Add the following apps to your INSTALLED_APPS:
 - `INSTALLED_APPS = [`
 - `# Default Django apps`
 - `'django.contrib.admin',`
 - `'django.contrib.auth',`
 - `'django.contrib.contenttypes',`
-

- 'django.contrib.sessions',
 - 'django.contrib.messages',
 - 'django.contrib.staticfiles',
 - # Third-party apps
 - 'allauth',
 - 'allauth.account',
 - 'allauth.socialaccount',
 - # Social providers
 - 'allauth.socialaccount.providers.google',
 - 'allauth.socialaccount.providers.facebook',
 - 'allauth.socialaccount.providers.github',
 -]
 - **Update Authentication Backends**
 - Update AUTHENTICATION_BACKENDS to include allauth backends.
 - AUTHENTICATION_BACKENDS = [- 'django.contrib.auth.backends.ModelBackend',
 - 'allauth.account.auth_backends.AuthenticationBackend',]
-



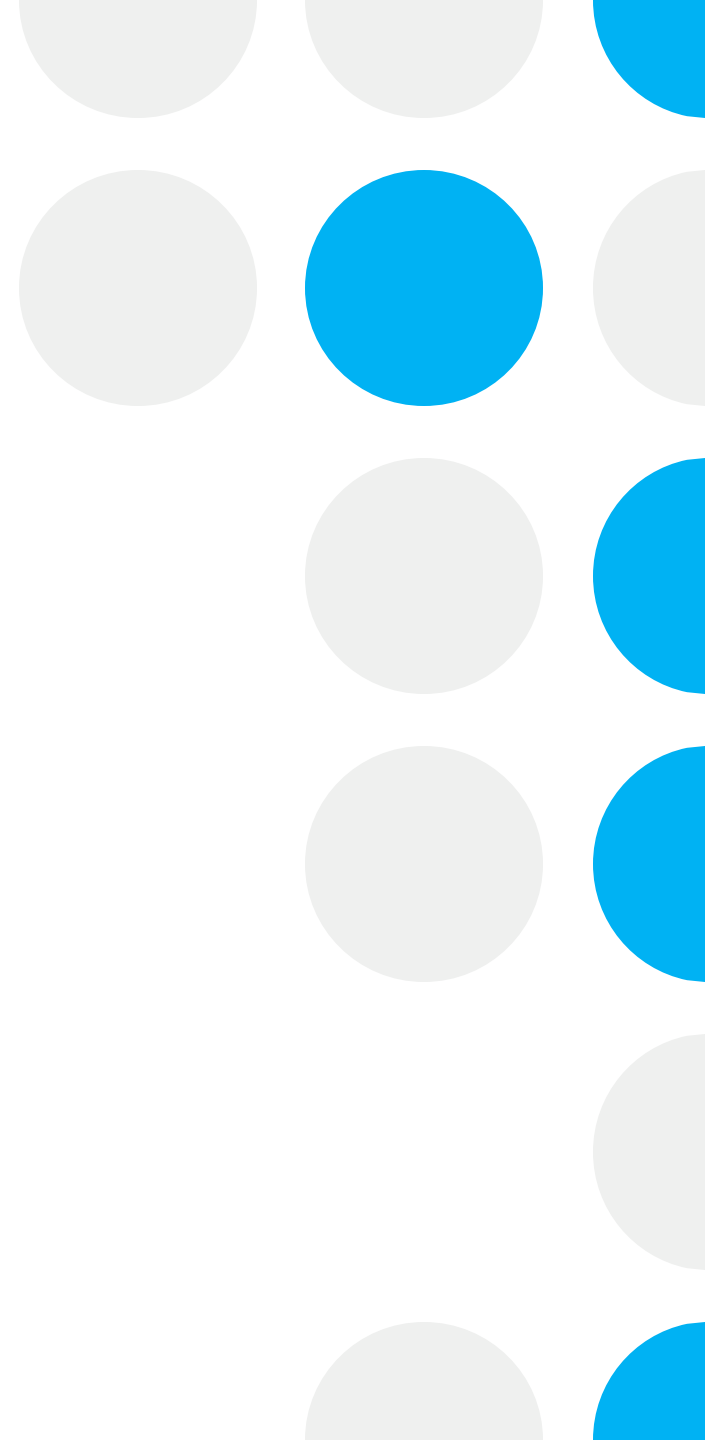
➤ **Configure URL Patterns**

- Include django-allauth URLs in your urls.py.
- `from django.contrib import admin`
- `from django.urls import path, include`
- `urlpatterns = [`
- `path('admin/', admin.site.urls),`
- `path('accounts/', include('allauth.urls'))`, # Allauth URLs
- `]`

➤ **Configure Settings for Allauth**

- Add the following configuration to your settings.py:
 - # Allauth settings
 - `LOGIN_REDIRECT_URL = '/'` # Redirect after login
 - `LOGOUT_REDIRECT_URL = '/'` # Redirect after logout
 - `ACCOUNT_AUTHENTICATION_METHOD = 'username_email'`
 - `ACCOUNT_EMAIL_REQUIRED = True`
 - `ACCOUNT_USERNAME_REQUIRED = True`
 - `SOCIALACCOUNT_QUERY_EMAIL = True`
-

- **Set Up Social Providers**
 - Go to the Django admin panel (/admin) and navigate to **Sites** under **django.contrib.sites**.
 - Update the default site to your domain (or localhost during development).
 - **Google**
 - Go to Google Cloud Console.
 - Create a new project and enable the **"OAuth consent screen"**.
 - Configure the OAuth 2.0 credentials:
 - Application type: **Web Application**.
 - Redirect URI:
<http://localhost:8000/accounts/google/login/callback/>.
 - Copy the **Client ID** and **Client Secret**.
 - In the Django admin panel:
 - Go to **Social Applications**.
 - Add a new application for Google:
 - Provider: **Google**
 - Name: **Google Login**
 - Client ID and Secret: Paste the values from Google.
 - Sites: Select your site.
-



➤ Facebook

- Go to [Facebook for Developers](#).
- Create a new app and set up **Facebook Login**.
- Configure the OAuth redirect URI:
 - <http://localhost:8000/accounts/facebook/login/callback/>
- Copy the **App ID** and **App Secret**.
- In the Django admin panel:
 - Add a new application for Facebook with the above credentials.

➤ GitHub

- Go to [GitHub Developer Settings](#).
 - Create a new OAuth application:
 - Authorization callback URL:
<http://localhost:8000/accounts/github/login/callback/>.
 - Copy the **Client ID** and **Client Secret**.
 - In the Django admin panel:
 - Add a new application for GitHub with the above credentials.
-

➤ **Test the Social Login**

- Run the Django development server:
- `python manage.py runserver`
- Visit `/accounts/login/` to see the social login options.
- Try logging in using Google, Facebook, or GitHub.

