

Neethu Babu

@neethubabu212

Why Your Django Projects Fail to Scale: Fix Your Folder Structure Today!

The perfect Django project folder explained in 8 simple steps.

Let's Swipe Right



Does this sound familiar?

- Spending hours hunting for the right file?
- Confused by unorganized app folders?
- Struggling to onboard new team members?

Let's Swipe Right



Clean Code Starts with Clean Folders.

A well-structured Django
project:

- Saves time.
- Reduces debugging nightmares.
- Scales effortlessly with your team.



Start with the Basics

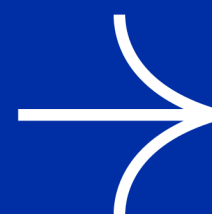
Essentials in your root folder:

- **manage.py**: Command center for your project.
- **requirements.txt**: Tracks dependencies.
- **.env**: Hides sensitive credentials.



Where the Magic Happens

- **settings.py**: Split into base, dev, and prod.
- **urls.py**: Maps routes to views.
- **wsgi.py / asgi.py**: Handles deployment entry points
- **__init__.py**: Don't skip it—this makes the folder a Python package.



Apps Keep It Organized

Every feature gets its own app.

Typical structure:

- **models.py**: Define your database schema.
- **views.py**: Manage logic for each request.
- **urls.py**: App-specific routes.
- **tests.py**: Test early. Test often.



Frontend, Meet Backend

- **Templates:** Keep them app-specific. Example: `templates/users/login.html`.
- **Static Files:** Organize into folders: `static/css/`, `static/js/`, etc.
- Use Django's static and template tags to link them correctly.



Find this useful?

A clean project structure is the first step to Django mastery.
Have your own tips? Share them in the comments!"

