

dj

Professionally setup django project



Created by **Ashish**



Why Proper Django Project Setup is Crucial?

Proper Django project setup is crucial for maintaining a structured and organized development environment, ensuring scalability, ease of collaboration, and efficient debugging throughout the software development lifecycle.



Chapter 1: Getting Started

1. Creating a Project Folder

Create a new directory to serve as the container for the Django project.

2. Open the Folder in VS Code

Use Visual Studio Code as the recommended code editor for the project.

Chapter 2:

Virtual Env

Install, create and activate a virtual environment

1. Installing virtualenv

```
pip install virtualenv
```

2. Creating virtual env

```
py -m venv venv
```

3. Activating virtual env

```
cd venv\scripts ; ./activate
```



Chapter 3:

Django Project

Use the command below and create a fresh project.

1. Installing Django in venv

```
pip install django
```

2. Creating Django Project

```
django-admin startproject  
<project name>
```

3. Creating Django App

```
py manage.py startapp  
<app name>
```

Chapter 4:

Static &

Template Files

1. Creating a Static Folder

Establish a folder for static files (e.g., stylesheets, scripts, images).

2. Creating a Templates Folder

Create a folder for HTML templates such as `base.html`, `index.html`...

Chapter 5: Configuring Settings.py

1. Adding Installed Apps

```
INSTALLED_APPS = [  
    <app_name>  
]
```

2. Configure static files

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR,  
        'static')]
```

3. Configure templates

```
'DIRS':  
[os.path.join(BASE_DIR,  
    'templates')],
```

Chapter 6:

Configuring

urls.py

1. Adding Installed Apps

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

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


Chapter 6: Configuring urls.py

2. Creating urls.py for the app

```
from django.urls import
path
from <your_app_name> import
views

urlpatterns = [
    path('', views.index,
name='home'),
    path('about/',
views.about, name='about'),
    path('contact/',
]
```

Chapter 7:

Creating

views.py

1. Creating functions to render pages

```
from django.shortcuts
import render

# Home page
def index(request):
    return render(req
'index.html', {})

# About page
def about(request):
    return render(req
'about.html', {})
```



Chapter 8: Migrations

1. Running Migrations for Database Setup

```
py manage.py makemigrations
```

```
py manage.py migrate
```

Chapter 9:

Superuser

1. Setting Up Admin Access

```
python manage.py  
createsuperuser
```

Chapter 10: Running

1. Running project on local host

```
python manage.py  
runserver
```



Chapter 11: Conclusion

In summary, this guide has equipped you with a structured approach to kickstart Django projects, emphasizing organization. Feel free to customize and expand for specific project needs.

dj

Download the project & start editing.



[github profile](#)

[linkedin profile](#)

[django-setup](#)

