Git and Django Setup Guide

1. Git Configuration

git init

Initializes an empty Git repository in your current folder so Git can track changes.

git config --global user.name "Your Name"

Sets your name globally for Git commits.

git config --global user.email "your@email.com"

Sets your email for all Git commits.

2. Clone GitHub Repo

git clone https://github.com/username/repo.git

Downloads a project from GitHub to your local machine.

3. Create & Activate Python Virtual Environment

py -m venv myvenv

Creates a virtual environment named myvenv (isolates your Django and other package installations).

myvenv\Scripts\activate

Activates the virtual environment on Windows.

4. Navigate and Install Packages

cd demo

Changes directory to demo, your Django project folder.

pip list

Shows all installed Python packages.

5. Install Django and Pillow

pip install django

Git and Django Setup Guide

Installs Django framework.

pip install pillow

Installs Pillow (used for image uploading/processing in Django).

6. Freeze Requirements

pip freeze > requirement.txt

Creates a requirement.txt file listing all packages. Useful when sharing the project or deploying.

7. Git Workflow

git status

Shows changes in the working directory and staging area.

git add .

Stages all changed files (adds them to be committed).

git commit -m "add change"

Saves the staged changes in Git with a message.

git push

Pushes your committed changes to GitHub (only works after setting remote).

8. Django Project Setup

django-admin startproject project .

Creates a new Django project in the current folder (. = no extra directory).

Folder structure:

```
project/
 manage.py
 project/
    settings.py
```

urls.py

etc.

9. Run Django Server

python manage.py runserver

Starts Django development server at:

http://127.0.0.1:8000/

10. Create Django Apps

python manage.py startapp polls

Creates a new app called polls. Apps are like modules for different features (e.g., blog, users, shop).

Creates:

polls/

admin.py

models.py

views.py

etc.

11. Start Another App (Users)

python manage.py startapp users

Creates another app, e.g., to manage user registration/login.

OR: python manage.py startapp apps

Creates a generic app named apps.

You can create many apps inside one Django project, each handling specific features.

Summary Table:

Command	Description
git init	Start Git tracking

Git and Django Setup Guide

py -m venv myvenv Create virtual environment

myvenv\Scripts\activate Activate it

pip install django Install Django

django-admin startproject project . Create Django project

python manage.py startapp appname Create Django app

python manage.py runserver Start dev server

git add . / git commit -m / git push Save and upload code to GitHub