**Django Notes App: Full Beginner Guide**

**🎓 What You'll Build**

A simple Notes app with Django where users can:

* View all notes
* Add a new note
* Edit a note
* Delete a note

**✅ Step 1: Install Django**

pip install django

**✅ Step 2: Start a New Django Project**

django-admin startproject notes\_project

cd notes\_project

**✅ Step 3: Create a Django App**

python manage.py startapp notes\_app

Add your app to INSTALLED\_APPS in notes\_project/settings.py:

INSTALLED\_APPS = [

...,

'notes\_app',

]

**✅ Step 4: Create the Note Model**

In notes\_app/models.py:

from django.db import models

class Note(models.Model):

title = models.CharField(max\_length=200)

content = models.TextField()

created\_at = models.DateTimeField(auto\_now\_add=True)

def \_\_str\_\_(self):

return self.title

Run migrations:

python manage.py makemigrations

python manage.py migrate

**✅ Step 5: Set Up the Admin**

In notes\_app/admin.py:

from django.contrib import admin

from .models import Note

admin.site.register(Note)

Create a superuser:

python manage.py createsuperuser

Run the server:

python manage.py runserver

Visit /admin and log in to add some notes.

**✅ Step 6: Create URLs and Views**

In notes\_app/urls.py:

from django.urls import path

from . import views

urlpatterns = [

path('', views.index, name='index'),

path('add/', views.add\_note, name='add\_note'),

path('edit/<int:note\_id>/', views.edit\_note, name='edit\_note'),

path('delete/<int:note\_id>/', views.delete\_note, name='delete\_note'),

]

Link app URLs in notes\_project/urls.py:

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('notes\_app.urls')),

]

**✅ Step 7: Create Views**

In notes\_app/views.py:

from django.shortcuts import render, redirect

from .models import Note

def index(request):

notes = Note.objects.all().order\_by('-created\_at')

return render(request, 'notes\_app/index.html', {'notes': notes})

def add\_note(request):

if request.method == 'POST':

Note.objects.create(

title=request.POST['title'],

content=request.POST['content']

)

return redirect('index')

return render(request, 'notes\_app/add\_note.html')

def edit\_note(request, note\_id):

note = Note.objects.get(id=note\_id)

if request.method == 'POST':

note.title = request.POST['title']

note.content = request.POST['content']

note.save()

return redirect('index')

return render(request, 'notes\_app/edit\_note.html', {'note': note})

def delete\_note(request, note\_id):

Note.objects.get(id=note\_id).delete()

return redirect('index')

**✅ Step 8: Create Templates**

Folder structure:

notes\_app/

├── templates/

└── notes\_app/

├── index.html

├── add\_note.html

└── edit\_note.html

**index.html**

<h1>📒 My Notes</h1>

<a href="{% url 'add\_note' %}">➕ Add Note</a>

{% for note in notes %}

<div>

<h3>{{ note.title }}</h3>

<p>{{ note.content }}</p>

<small>{{ note.created\_at }}</small><br>

<a href="{% url 'edit\_note' note.id %}">✏️ Edit</a>

<a href="{% url 'delete\_note' note.id %}" onclick="return confirm('Delete this note?');">🗑️ Delete</a>

</div>

{% empty %}

<p>No notes found.</p>

{% endfor %}

**add\_note.html**

<h1>➕ Add Note</h1>

<form method="POST">{% csrf\_token %}

Title: <input type="text" name="title"><br>

Content: <textarea name="content"></textarea><br>

<button type="submit">Save</button>

</form>

<a href="{% url 'index' %}">← Back</a>

**edit\_note.html**

<h1>✏️ Edit Note</h1>

<form method="POST">{% csrf\_token %}

Title: <input type="text" name="title" value="{{ note.title }}"><br>

Content: <textarea name="content">{{ note.content }}</textarea><br>

<button type="submit">Update</button>

</form>

<a href="{% url 'index' %}">← Back</a>

**✅ Step 9: Deploy to Render**

* Create requirements.txt:

pip freeze > requirements.txt

* Install gunicorn:

pip install gunicorn

* Create render.yaml with deploy config
* Push to GitHub
* Connect GitHub repo on [render.com](https://render.com/)