Django File upload:

**1. Create a Django Project and App**

First, create a Django project and an app within the project:

django-admin startproject file\_upload\_project

cd file\_upload\_project

python manage.py startapp file\_upload\_app

**2. Define Model for File Upload**

In `file\_upload\_app/models.py`, define a model to store uploaded files:

from django.db import models

class UploadedFile(models.Model):

file = models.FileField(upload\_to='uploads/')

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

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

return self.file.name

**3. Create Form for File Upload**

In `file\_upload\_app/forms.py`, create a form for uploading files:

from django import forms

from .models import UploadedFile

class UploadFileForm(forms.ModelForm):

class Meta:

model = UploadedFile

fields = ['file']

**4. Create Views for File Upload**

In `file\_upload\_app/views.py`, create views for handling file upload:

from django.shortcuts import render, redirect

from .forms import UploadFileForm

def upload\_file(request):

if request.method == 'POST':

form = UploadFileForm(request.POST, request.FILES)

if form.is\_valid():

form.save()

return redirect('upload\_success')

else:

form = UploadFileForm()

return render(request, 'file\_upload\_app/upload.html', {'form': form})

def upload\_success(request):

return render(request, 'file\_upload\_app/upload\_success.html')

**5. Create HTML Templates**

Create HTML templates for the upload form and success page:

\*\*`file\_upload\_app/templates/file\_upload\_app/upload.html`:\*\*

<!DOCTYPE html>

<html>

<head>

<title>File Upload</title>

</head>

<body>

<h2>Upload File</h2>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

</body>

</html>

\*\*`file\_upload\_app/templates/file\_upload\_app/upload\_success.html`:\*\*

<!DOCTYPE html>

<html>

<head>

<title>Upload Success</title>

</head>

<body>

<h2>File Uploaded Successfully!</h2>

<a href="{% url 'upload\_file' %}">Upload Another File</a>

</body>

</html>

**6. Configure URLs**

Configure URLs in `file\_upload\_app/urls.py`:

from django.urls import path

from . import views

urlpatterns = [

path('upload/', views.upload\_file, name='upload\_file'),

path('success/', views.upload\_success, name='upload\_success'),

]

**7. Include App URLs in Project URLs**

Include app URLs in the project's `urls.py`:

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

path('', include('file\_upload\_app.urls')),

]

**Explanation:**

- \*\*Model\*\*:

`UploadedFile` model stores uploaded files with a `FileField` for the file itself and a `DateTimeField` for the upload time.

- \*\*Form\*\*:

`UploadFileForm` is a model form for the `UploadedFile` model.

- \*\*Views\*\*:

`upload\_file` handles file upload, rendering the upload form and processing POST requests. `upload\_success` displays a success message after a successful upload.

- \*\*Templates\*\*:

`upload.html` contains the form for file upload, while `upload\_success.html` displays a success message.

- \*\*URLs\*\*:

URLs are configured to map to the `upload\_file` and `upload\_success` views.

- \*\*Settings\*\*:

Make sure to configure `MEDIA\_ROOT` and `MEDIA\_URL` settings in `settings.py` to specify where uploaded files are stored and served from.

That's it! With this setup, users can upload files using the provided form, and the uploaded files will be saved in the specified directory.