How to Upload Multiple Files in Django: Images, Files, and PDF

A feature that is essential for many online applications is the ability to allow users to upload many files and easily display them on a single page. Enabling multiple file uploads and seamless display improves user experience and functionality, regardless of whether you're developing a media sharing website, collaboration platform, or any other application involving file interaction. In this complete tutorial, we'll walk you through implementing this functionality in a Django web application while incorporating the changes you've supplied.

Installation and Setup of Django

```
pip install django
pip install Pillow
```

Step 1: Setting Up Your Django Project

```
django-admin startproject multipleImage
```

Change directory

```
cd multipleImage
python manage.py startapp first_app
```

Step 2: Create a Model first_app app models.py

```
class UploadImage(models.Model):
    file = models.FileField(upload_to='account/images/')
```

Step 3: Include the UploadImage model in the admin.py file for the first_app app.

```
from .models import UploadImage
admin.site.register(UploadImage)
```

Step 4: Configure Media Settings settings.py

```
INSTALLED_APPS = [
    'first_app', # add this
]
# Media configuration
import os
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

Step 5: In first_app, creating forms.py

```
from django import forms
from .models import UploadImage

class UploadImageForm(forms.ModelForm):
    class Meta:
      model = UploadImage
      fields = ['file']
```

Step 5: In first_app, creating views.py

```
from django.shortcuts import render, redirect
from .forms import UploadImageForm
from .models import UploadImage
from django.views import View

def UploadImageView(request):
    files = UploadImage.objects.all()
    form = UploadImageForm(request.POST, request.FILES)

    if form.is_valid():
      img = request.FILES.getlist('file')
      for i in img:
        UploadImage.objects.create(file=i)
      return redirect('home')

    return render(request, 'index.html', {'form':form, 'files':files})
```

Step 6: Open the (multipleImage/urls.py) and add the following URL:

```
from django.contrib import admin
from django.conf import settings # new
from django.conf.urls.static import static # new
from django.urls import path
```

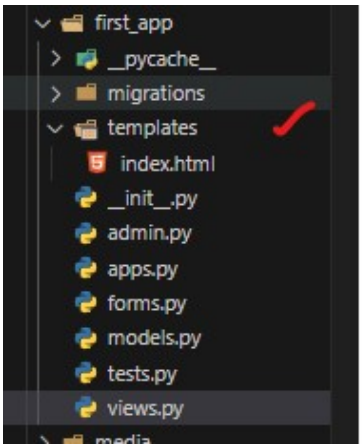
```
from first_app.views import UploadImageView # new

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', UploadImageView, name='home'), # new
]
if settings.DEBUG: urlpatterns += static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

Step 7: Running Migrations

```
python manage.py makemigrations
python manage.py migrate
```

Step 8: In first_app, Creating Templates folder




```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Upload and Display Files</title>
</head>

<body>
  <h1 style="text-align: center;">Upload and Display Images</h1>
  <div style="display: flex; justify-content:center;">

    <form method="post" enctype="multipart/form-data">
      {% csrf_token %}
      <label for="{{ form.file.id_for_label }}">File</label>
      <input type="file" multiple name="{{ form.file.name }}"><br>
      <button style="margin-top: 5px;" type="submit">submit</button>
```

```
    </form>
  </div>
  <div>
    {% for i in files %}
    <img src={{i.file.url}} alt="">
    {% empty %}
    <li>No files uploaded.</li>
    {% endfor %}
  </div>
</body>

</html>
```

Step 9: Now, start the server

```
python manage.py runserver
```

In your browser, go to http://127.0.0.1:8000/ to visit the page where you may upload and view files at the same time.

Conclusion

Best wishes! You've successfully integrated the Django application's ability to accept multiple file uploads and display them all on the same page. You have created the fundamental elements—models, views, forms, templates, and configurations—necessary to develop an interactive and user-friendly file management functionality by following this thorough tutorial. Comment below to suggest further tutorials of this kind.

You may find this project on Github.