TRAINING DAY12 REPORT

06 JULY 2025

DJANGO FORMS

Django Forms provide a powerful way to **handle user input**, **render HTML forms**, **validate data**, and **save to the database** — all in a clean and secure way.

Types of Forms in Django

Type	Description
forms.Form	Used for creating forms **not directly tied** to models (manual field definition).
forms.ModelForm	Tied to a Django model — automatically creates form fields based on the model.

Example 1: Using forms.Form

1. forms.py

```
blogapi > blog > 🐍 forms.py > ...
1    from django import forms
2
3    class ContactForm(forms.Form):
4        name = forms.CharField(max_length=100)
5        email = forms.EmailField()
6        message = forms.CharField(widget=forms.Textarea)
7
8
```

2. views.py

```python
from django.shortcuts import render
from .forms import ContactForm

def contact_view(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            # process form.cleaned_data['name'], etc.
            return render(request, 'success.html')
    else:
        form = ContactForm()

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

3. contact.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Send</button>
    </form>

</body>
</html>
```

Example 2: Using forms.ModelForm

1. models.py

```python
blogapi > blog > models.py > Product
1   from django.db import models
2
3   class Student(models.Model):
4       name = models.CharField(max_length=100)
5       roll_no = models.IntegerField()
6       email = models.EmailField()
7
8
```

2. forms.py

```python
blogapi > blog > forms.py > ...
1   from django import forms
2   from .models import Student
3
4   class StudentForm(forms.ModelForm):
5       class Meta:
6           model = Student
7           fields = ['name', 'roll_no', 'email']
8
```

3. views.py

```python
blogapi > blog > views.py > ...
1   from .forms import StudentForm
2
3   def register_student(request):
4       if request.method == 'POST':
5           form = StudentForm(request.POST)
6           if form.is_valid():
7               form.save()
8               return render(request, 'success.html')
9       else:
10          form = StudentForm()
11      return render(request, 'register.html', {'form': form})
12
```

Useful Form Methods & Attributes

Feature	Description
form.is_valid()	Checks if submitted data meets field requirements
form.cleaned_data	Returns a dictionary of validated data
form.errors	Displays validation errors
form.save()	Saves the form (ModelForm only)
form.as_p, as_table, as_ul	Renders form fields in different HTML formats

CSRF Protection

Always include:

{% csrf_token %}

inside your <form> tag to protect against CSRF attacks.

FILE AND IMAGE UPLOADS

How Django Handles Uploads

- Django stores uploaded files in a location defined in settings.py (MEDIA_ROOT).

- The **file's path** is stored in the database, **not the file's content**.

- Requires enctype="multipart/form-data" in form tag.

1. **In settings.py:**

```python
import os

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

2. **In urls.py:**

```python
blogapi > blogapi > 🐍 urls.py > ...
17
18    from django.conf import settings
19    from django.conf.urls.static import static
20
21    urlpatterns = [
22        # ... your other urls
23    ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
24
25
```

3. **Models for File & Image Upload**

File Upload Model

```python
blogapi > blog > 🐍 models.py > ...
1    from django.db import models
2
3    class Document(models.Model):
4        title = models.CharField(max_length=100)
5        uploaded_file = models.FileField(upload_to='documents/')
6
7
```

Image Upload Model

```
blogapi > blog > 🐍 models.py > ...
1    class Photo(models.Model):
2        caption = models.CharField(max_length=100)
3        image = models.ImageField(upload_to='images/')
4
5
```

NOTE: For imageField , install pillow.

4. **Forms**

```
blogapi > blog > 🐍 forms.py > ...
1    from django import forms
2    from .models import Document, Photo
3
4    class DocumentForm(forms.ModelForm):
5        class Meta:
6            model = Document
7            fields = ['title', 'uploaded_file']
8
9    class PhotoForm(forms.ModelForm):
10       class Meta:
11           model = Photo
12           fields = ['caption', 'image']
13
```

5. Views

```
blogapi > blog > 🐍 views.py > ...
  1    from django.shortcuts import render, redirect
  2    from .forms import DocumentForm, PhotoForm
  3
  4    def upload_file(request):
  5        if request.method == 'POST':
  6            form = DocumentForm(request.POST, request.FILES)
  7            if form.is_valid():
  8                form.save()
  9                return redirect('success')
 10        else:
 11            form = DocumentForm()
 12        return render(request, 'upload_file.html', {'form': form})
 13
 14    def upload_image(request):
 15        if request.method == 'POST':
 16            form = PhotoForm(request.POST, request.FILES)
 17            if form.is_valid():
 18                form.save()
 19                return redirect('success')
 20        else:
 21            form = PhotoForm()
 22        return render(request, 'upload_image.html', {'form': form})
 23
```

6. HTML Templates

File Upload (upload_file.html)

```html
templates > 5 contact.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <form method="post" enctype="multipart/form-data">
10         {% csrf_token %}
11         {{ form.as_p }}
12         <button type="submit">Upload File</button>
13     </form>
14
15
16 </body>
17 </html>
18
```

Image Upload (upload_image.html)

```html
templates > 5 contact.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <form method="post" enctype="multipart/form-data">
10         {% csrf_token %}
11         {{ form.as_p }}
12         <button type="submit">Upload Image</button>
13     </form>
14
15
16
17 </body>
18 </html>
19
```

Displaying Uploaded Files & Images

Display files:

```
<a href="{{ document.uploaded_file.url }}">{{ document.uploaded_file.name }}</a>
```

Display images:

```
<img src="{{ photo.image.url }}" alt="{{ photo.caption }}">
```