Django Forms

❓ What are Django Forms?

Django Forms handle **user input** and **validation**. They help in:

- Rendering HTML forms
- Validating data
- Saving data to the database (when using ModelForms)

✅ Types of Forms

1. **forms.Form** – for custom, non-model-based forms
2. **forms.ModelForm** – for forms tied directly to Django models

```
from django import forms

class ContactForm(forms.Form):
    name = forms.CharField(max_length=100)
    email = forms.EmailField()
    message = forms.CharField(widget=forms.Textarea)
```

📥 Using the Form in a View

```
python

def contact_view(request):
    form = ContactForm(request.POST or None)
    if form.is_valid():
        # process data
        ...
    return render(request, 'contact.html', {'form': form})
```

🖋 HTML Template

```
html

<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Send</button>
</form>
```

- {{ form.as_p }} – renders fields as <p> tags
- Alternatives: as_table, as_ul

🏭 ModelForm Example

```
python

from .models import Student

class StudentForm(forms.ModelForm):
    class Meta:
        model = Student
        fields = ['name', 'age', 'email']
```

🔧 Customizing Fields

```
python

name = forms.CharField(label='Full Name',
widget=forms.TextInput(attrs={'class': 'form-control'}))
```

🛡 Form Validation

- **clean_<field>()** – validate individual fields
- **clean()** – validate entire form

```
python

def clean_email(self):
    email = self.cleaned_data['email']
    if not email.endswith('@example.com'):
        raise forms.ValidationError("Must use example.com
email")
    return email
```

📤 Saving Form Data

```
python

if form.is_valid():
    form.save()  # For ModelForm
```

🔄 Initial Data

```
python

form = ContactForm(initial={'name': 'John Doe'})
```

📚 Useful Tips

- Always use request.POST for POST data and request.GET for GET.