

Forth Assignment

Thuraya Waheeb Al-hobaishi

Supervision of Mr.Malek

2025/8/22

مقارنة بين Form Django و Form HTML

الجانب	HTML Form	Django Form
المكان	يُكتب داخل صفحة HTML (Front-End)	يُكتب في ملف forms.py (Back-End)
الوظيفة	يجمع بيانات من المستخدم ويرسلها للسيرفر	يستقبل البيانات من الـ HTML Form ويعالجها
التحقق	محدود (required, minlength...) وغالباً يحتاج JavaScript	مدمج وقوي (is_valid(), cleaned_data)
الأمان	ما يحمي من الاختراق (SQL Injection, XSS)	فيه حماية جاهزة (CSRF, sanitization, validation)
الارتباط بالبيانات	ما له علاقة مباشرة بقاعدة البيانات	ممکن يكون مرتبط بموديل عبر ModelForm
طريقة العرض	لازم تكتب كل الحقول يدوياً بـ <input> و <textarea>	ممکن تخلي Django يولدها تلقائياً ({{ form.as_p }})
المرونة	حرية كاملة بالتصميم والشكل	يركز على المنطق ومعالجة البيانات، والشكل يُعدل بالـ template

مثال للتوضيح:

HTML Form عادي:

```
1 <form action="/login/" method="post">
2   <input type="text" name="username" placeholder="Enter username">
3   <input type="password" name="password" placeholder="Enter password">
4   <button type="submit">Login</button>
5 </form>
```

Django Form:

```
1 from django import forms
2
3 class LoginForm(forms.Form):
4     username = forms.CharField(max_length=100)
5     password = forms.CharField(widget=forms.PasswordInput)
6
```

:views.py



```
1  from django.shortcuts import render
2  from .forms import LoginForm
3
4  def login_view(request):
5      if request.method == "POST":
6          form = LoginForm(request.POST)
7          if form.is_valid():
8              # معالجة البيانات بشكل آمن
9              print(form.cleaned_data["username"])
10     else:
11         form = LoginForm()
12     return render(request, "login.html", {"form": form})
13
```

:template (login.html)



```
1  <form method="post">
2      {% csrf_token %}
3      {{ form.as_p }}
4      <button type="submit">Login</button>
5  </form>
```

HTML Form

Login

Username

Password

Login

Django Form

Login

Username

Password

Login