علی حیدری

سوال ۱ : در جنگو، ویو یک تابع یا کلاس پایتونی است که یک درخواست (request) را می‌گیرد و یک پاسخ (response)برمی‌گرداند. ویو را می‌توان لایه‌ی میانی بین داده‌ها و تمپلیت ها دانست.

سوال ۲: تفاوت ویوی تابعی (FBV) و ویوی کلاسی(CBV)

ویوی تابعی:(FBV)

به صورت یک تابع پایتون نوشته می‌شود.

ساده و مستقیم است و راحت فهمیده می‌شود.

ویوی کلاسی:(CBV)

به صورت یک کلاس نوشته می‌شود که از کلاس‌های آماده‌ی جنگو مثل View ارث‌بری می‌کند.

متدهایی مثل getو postدرخواست‌های مختلف را مدیریت می‌کنند.

برای پروژه‌های بزرگ‌تر، انعطاف‌پذیرتر و قابل استفاده مجدد است.

سوال۳ :آرگومان‌ها و خروجی یک ویوی تابعی

آرگومان‌ها:

همیشه حداقل یک آرگومان می‌گیرد : request شیHttpRequest .

می‌تواند آرگومان‌های اضافه هم بگیرد مثلاً id, slugکه در urls.pyتعریف می‌شوند.

خروجی:

باید یک HttpResponseیا زیرکلاس آن مثل JsonResponse یا HttpResponseRedirectبرگرداند.

سوال ۴: ویو چگونه پاسخ برمی‌گرداند؟

محتوای:HTML

با HttpResponse("<h1>Hello</h1>")

یا رایج‌تر با render(request, "template.html", context)

پاسخ:JSON

با JsonResponse({"key": "value"})

سوال۵: تفاوت HttpResponse و()render

HttpResponse:

پایه‌ای‌ترین کلاس پاسخ. وقتی می‌خواهیم داده‌ی خام HTML ساده، متن، XMLو ... بفرستید.

render ():

یک میان‌بر است:

قالب HTML را بارگذاری می‌کند.

داده‌ها را داخل آن قرار می‌دهد.

نتیجه را در قالب یک HttpResponse برمی‌گرداند.

چه زمانی استفاده شود؟

HttpResponse:برای خروجی‌های ساده یا داده‌های خام.

render ():برای ساخت صفحه‌ی کامل با قالب.

سوال ۶ : نقش تمپلیت ها در جنگو

تمپلیت ها مشخص می‌کنند داده‌ها چگونه به کاربر نمایش داده شوند.

معمولاً فایل‌های HTML هستند با جای‌خالی برای داده‌های پویا.

این باعث جداسازی منطق (views) از نمایش (templates) می‌شود.

سوال ۷ : چطور یک تمپلیت را بسازیم و به ویو وصل کنیم؟

یک فایل تمپلیت در پوشه‌ی templates/ بسازید.

در ویو از render()برای بارگذاری تمپلیت استفاده کنید.

در settings.pyمسیر تمپلیت ها مشخص می‌شود (معمولاً از قبل تنظیم شده).

سوال ۸:

Contextیک دیکشنری از داده‌هاست که ویو به تمپلیت می‌فرستد.

متغیرهای داخل آن در تمپلیت با {{ variable_name }}قابل دسترسی هستند.

سوال ۹ : تگ‌ها و فیلترهای تمپلیت

Template Tags:

دستوراتی داخل {% ... %}برای کنترل منطق در تمپلیت(مثل حلقه‌ها و شرط‌ها).

Template Filters:

تغییر یا پردازش متغیرها داخل {{ }}

مثال{{ name|upper }} :

سوال ۱۰: ارث‌بری تمپلیت

برای جلوگیری از تکرار کدهای ثابت مثل ,<head>منو، فوتر یک فایل پایه (base.html) می‌سازیم.

قالب‌های دیگر با {% extends "base.html" %}از آن ارث‌بری می‌کنند.

بخش‌های قابل تغییر در {% block %} ... {% endblock %}قرار می‌گیرند.

سوال ۱۱ : نقش فایلurls.py

URL dispatcherاست: مسیر URL را به ویوی مناسب وصل می‌کند.

کاربر آدرس را وارد می‌کند -> جنگو بررسی می‌کند -> ویو اجرا می‌شود -> پاسخ داده می‌شود.

سوال ۱۲ : تابعpath()

برای تعریف الگوی URL استفاده می‌شود.

Route: مسیر url

View: تابع یا کلاس مربوطه.

Name: نام اختیاری برای ارجاع در جاهای دیگر.

مثل :

path("article/<int:id>/", views.article_detail, name="article_detail")

اگر کاربر /article/10/را بزند -> تابع article_detail(request, id=10)اجرا می‌شود.

سوال ۱۳: نام‌گذاری الگوهای URL و اهمیت آن

با آرگومان name به یک مسیر URL اسم می‌دهیم.

مزیت:

به جای اینکه در تمپلیت ها آدرس مستقیم بنویسیم، از اسم استفاده می‌کنیم.

اگر URL تغییر کند، فقط کافیست در urls.py تغییر داده شود.