به نام خدا

علیرضا مختاری

پاسخ تسک اول

سوال ۱:

وب سرور وظیفه ی ذخیره سازی و ارسال محتوای یک وبسایت را به کلاینتی که آن را درخواست کرده است .محتوای درخواست شده میتواند هر چیزی باشد .رایج ترین نوع کلاینت مرورگرها هستند.

وب سرور یک بخش سخت افزاری و یک بخش نرم افزاری دارد. وب سرور یک رایانه است که نرم افزار وب سرور روی آن اجرا میشود و فایل های اجزای وبسایت را ذخیره میکند(مثل اسناد html و استایل شیت های css). وب سرور متصل به اینترنت است و اطلاعات با سایردستگاه های متصل تبادل اطلاعات میکند. از نظر نرم افزاری وب سرور شامل چندین بخش(حداقل یک سرور http) است که وظیفه ی کنترل چگونگی دسترسی کاربران به فایل های روی هاست را بر عهده دارد.

نحوه ی کارکرد : هر بار مرورگر به فایلی که در وب سرور موجود است نیاز پیدا میکند، مرورگر فایل را از طریق http درخواست میکند. این درخواست به دست وب سرور(سخت افزار) و سرور http (نرم افزار) درخواست را پذیرش میکند.در صورت پیدا کردن فایل مورد نیاز پاسخ درخواست از طریق http به مرورگر فرستاده میشود.
سرور http بخشی از نرم افزار وب سرور است که URL ها و http (پروتکلی که مرورگر برای مشاهده ی صفحات وب استفاده میکند) درک میکند.

نقش nginx در وب سرور: یکی از نرم افزارهایی است که روی وب سرور پیاده سازی میشود و وظیفه اش این است که درخواست های مرورگر را دریافت، پردازش و پاسخ بدهد که از طریق http اینکار را انجام میدهد و میتواند نقش های دیگری هم داشته باشد مثل: reverse proxy،load balancer و cache server.

سوال ۲:

اتصال میان مرورگر و سرور اینگونه است که ابتدا مرورگر(کلاینت) یک درخواست http به آدرس سرور میفرستد .این درخواست شامل متد (post,Get) و هدرها(اطلاعات اضافی مثل کوکی ها، نوع محتوا،توکن احراز هویت) و بدنه است.سرور پس از دریافت درخواست مرورگر درخواست را پردازش میکند و یک پاسخ آماده میکند و سپس سرور به مرورگر یک پاسخ http برمیگرداند و در نهایت مرورگر پاسخ را دریافت کرده و نمایش میدهد.

سوال ۳:

HTTP مخفف عبارت hyper text transfer protocol به معنای پروتکل انتقال فرامتن است.در واقع مجموعه ای از قوانین برای انتقال فایل ها از جمله فایل های متنی، عکس، صدا و تصویر و فایل های چند رسانه ای از طریق وب است. در پروتکل HTTP برای تبادل اطلاعات بین سرور و کلاینت، ابتدا درخواستی از مرورگر به وب سرور ارسال میشود،سرور باید به این درخواست پاسخ بدهد اگر اطلاعات درخواست شده توسط کلاینت موجود باشد، سرور پاسخی مبنی بر وجود اطلاعات و موافقت با ارسال آن ها، برای کاربر میفرستد.سپس کاربر درخواستی حاوی پیامی برای دریافت اطلاعات به سمت سرور ارسال کرده و سرور با دریافت این پیام، داده های درخواستی را به بسته های اطلاعاتی کوچک تری تقسیم کرده است و برای کاربران ارسال میکند.

وقتی یک آدرس URL را در مرورگر سرچ میکنیم، URL با استفاده از Dns Resolution تبدیل به آدرس IP میشود. مرورگر با استفاده از IP، از طریق پروتکل TCP به پورت مناسب متصل میشود و سپس مرورگر یک درخواست HTTP به سرور میفرستد و سرور این درخواست را پردازش میکند

سپس سرور پاسخ درخواست HTTP را به مرورگر برمیگرداند و سپس در مرورگر نمایش داده میشود.

سوال ۵,۴:

وقتی کاربر آدرس سایت را سرچ میکند، چندین مرحله وجود دارد تا درخواست به Django برسد.

Client: مرورگر یک درخواست http میفرستد.

Internet and network layer:درخواست از طریق اینترنت به وب سرور فرستاده میشود.

Web server: درخواست رسیده به توسط وب سرور بررسی میشود که آیا درخواست فایل استاتیکه(js,css,picture) در غیر اینصورت باید به برنامه ی بک اند برود و اگر بک اند باشد، از طریق WSGI یا ASGI به Django فرستاده میشود.

Application Server: این بخش رابط بین وب سرور و برنامه ی Djangoاست.درخواست HTTP را به یک شی پایتون قابل پردازش برای Django تبدیل میکند.

Django App(backend): منطق اصلی برنامه اجرا میشود، با دیتابیس،فایل ها و سرویس های اطلاعاتی دیگر کار میکند. پاسخ را ساخته و به Application Server میفرستد.

Response path: پاسخ Application Server را به وب سرور میفرستد و وب سرور پاسخ را به کلاینت(مرورگر) بازمیگرداند.

Browser ⟶ Internet ⟶ Web Server ⟶ Application Server ⟶ Django App ⟶ Data Base / File / API

به طور خلاصه Django پس از وب سرور و Application Server قرار دارد.

همچنین تفاوت Application Server و وب سرور این است که وب سرور برای محتوای استاتیک و ثابت استفاده میشود یعنی برای مثال اطلاعاتی که فارغ از اینکه که کلاینت در کدام موقعیت جغرافیایی قرار دارد نمایش داده میشود و Application Server برای ارسال محتوای داینامیک و قابل تغییر.

سوال ۶:

Django یک فریمورک برپایه‌ی زبان پایتون است که از الگوی معماری MVT(Model_View_Template) استفاده میکند.

تفاوت معماری MVT و MVC :

MVC(Model_view_Controller)

Model: لایه ای که داده ها و منطق دسترسی به پایگاه داده را مدیریت میکند.

View: لایه ای که داده ها را به کاربر نمایش میدهد.

Controller: واسط بین Model و View که درخواست ها را پردازش و پاسخ مناسب تولید میکند.

MVT(Model_View_Template)

Model: همان نقش مدل در MVC را ایفا میکند(مدیریت داده ها و تعامل با دیتابیس)

Template: به جای View در MVC وظیفه ی نمایش داده را برعهده دارد.

View: در Django همان نقش Controller را دارد یعنی منطق اصلی برنامه را اجرا میکند و داده را به عنوان Template میفرستد.

تفاوت اصلی میان این دو معماری این است که در MVT چیزی به اسم Controller جداگانه نداریم.View خودش همان نقش Controller را ایفا میکند و Django خودش مدیریت مسیریابی(URL Routing) را انجام میدهد.

نقش اجزای اصلی تشکیل دهنده ی Django:

URL: نقش مسیریابی را دارد. هر درخواستی که از مرورگر می آید بر اساس الگوهای تعریف شده در فایل به یک View خاص ارسال میشود.

View: این بخش همان جایی است که منطق اصلی برنامه نوشته میشود. به این صورت عمل میکند که درخواست ها را دریافت میکند و پس از پردازش یا گرفتن داده از Model داده ها را به Template ارسال میکند.

View معمولا یک تابع یا یک کلاس در فایل Views.py است.

Model: وظیفه ی این بخش تعریف ساختار داده و ارتباط با دیتابیس است. به صورت ORM(Object Relational Mapping) کار میکند یعنی با کد پایتون میتوان روی دیتابیس عملیات انجام داد.

Template: Template فایل های html هستندکه داده های ارسالی از View را نمایش میدهند. داخل Template میتوان از سینتکس مخصوص Django برای نمایش داده ها استفاده کرد.

سوال ۷:

Python virtual envirement یک محیط ایزوله برای هر پروژه پایتون است که نسخه ی اختصاصی مفسر و کتابخانه ها را جدا از سیستم اصلی نگه میدارد. این کار باعث میشود وابستگی ها و نسخه پکیج ها بین پروژه ها تداخل نداشته باشند.با فعالسازی آن، نصب و اجرای پکیج ها فقط در همان محیط انجام میشود و پروژه ها مستقل از هم عمل میکنند.

استفاده از محیط مجازی در Django یا هر پروژه پایتون مهم است چون وابستگی ها و نسخه ی کتابخانه ها را برای هر پروژه جدا نگه میدارد، از تداخل بین پروژه جلوگیری میکند و امکان اجرای پروژه روی محیط های مختلف با همان نسخه ها را فراهم میکند.

سوال ۸:

Django Rest Framework(DRF) برای ساخت API استفاده میشود که داده ها را در قالب هایی مانند JSON یا XML به کلاینت ارائه میدهد.این کلاینت ها معمولا برنامه های فرانت اند(مانند React یا Angular)، اپلیکیشن های موبایل یا سرویس های دیگر هستند.خروجی DRF داده ی خام است و برای زمانی استفاده میشود که نیاز داریم داده را به برنامه های دیگر بدهیم نه صفحات html.

در مقابل Django Templates برای ساخت صفحات html سمت سرور به کار می‌رود.در این روش خروجی به صورت HTML کامل تولید شده و مستقیما به مرورگر کاربر ارسال میشود. لین روش زمانی مناسب است که بخواهیم محتوای وب را بدون نیاز به به پردازش سمت کلاینت نمایش دهیم.

به طور خلاصه DRF مناسب ایجاد API برای تبادل داده بین سیستم ها است، در حالی که Django Templatesمناسب تولید و ارسال صفحات HTMLبرای مرورگر کاربر میباشد.

سوال ۹:

برای ساخت یک پروژه ی Django ابتدا باید python و Django را نصب کنیم و سپس مراحل زیر را اجرا کنیم:

ایجاد یک محیط مجازی(Virtual Envirement):

```
Python -m venv venv
```

فعالسازی محیط مجازی:

```
venv\scripts\activate
```

نصب Django:

```
pip install Django
```

ساخت پروژه جدید:

```
django-admin startproject projectname
```

اجرای سرور توسعه:

```
cd projectname
python manage.py runserver
```

ساختار پوشه‌های یک پروژه Django:

```
projectname/
    manage.py
    projectname/
        __init__.py
        settings.py
        urls.py
        asgi.py
        wsgi.py
```

سوال ۱۰:

برای ساخت اپ، از دستور زیر در مسیر پروژه استفاده میکنیم:

```
python manage.py startapp appname
```

سپس برای فعال کردن اپ، باید نام آن را در فایل settings.py و بخش INSTALLED_APPS اضافه کنیم:

```
INSTALLED_APPS = [
    ‘ appname’,
    . . .
]
```

ساختار پوشه های یک اپ به شکل زیر است:

```
appname/
    admin.py
    apps.py
    models.py
    tests.py
    views.py
    migrations/
```

هر اپ Django یک بخش مستقل از پروژه است که میتواند مدل ها، ویوها، قالب ها و فایل های خاص خود را داشته باشد. این اپ ها از طریق ثبت در INSTALLED_APPS به پروژه اصلی متصل میشوند.

what happens when visiting a URL

Request path(مسیر درخواست)

وقتی مرورگر آدرس سایت را باز میکند یک درخواست HTTP به سرور ارسال میشود.این درخوسات شامل اطلاعاتی مثل متد(GET/POST) هدرها و کوکی ها است.

URL Dispatcher

درخواست به Django میرسد و URL Dispatcher بررسی میکند که این URL به کدام view مرتبط است.این کار با فایل urls.py انجام میشود که مسیرها و View های مربوطه را مشخص کرده است.

View Handeling

پس از پیدا شدن View مربوطه تابع یا کلاس View اجرا میشود در این مثال View ما Hello World!My Name is Alireza Mokhtari است که یک پاسخ ساده Http Response ایجاد میکند.

Template Rendering

اگر View از Template استفاده کند، داده ها را به فایل HTML فرستاده شده و Django قالب را رندر میکند تا HTML نهایی تولید شود.در مثال ما، Template استفاده نشده و پاسخ مستقیم به مرورگر فرستاده میشود.

Response Path

پاسخ تولید شده توسط View به سرور برمیگردد و سرور آن را به مرورگر کلاینت ارسال مکینند.
مرورگر داده های دریافتی را نمایش میدهد و کاربر نتیجه را میبیند.