۱- وب سرور یک نرم‌افزار یا سخت‌افزاره که وظیفش اینه که درخواست کاربران از طریق مرورگر دریافت بکنه و پاسخشو ارسال به بکنه بهشون. معمولا هم این پاسخ ها شامل صفحات وب مثل جاوا اسکریپت و html، تصاویر، ویدیوها هستن.
Nginx یه وب سرور قدرتمنده که فایل ها یا محتوای وب به مرورگر کاربر به صورت استاتیک میفرسته و در کل وظیفه‌اش مدیریت درخواست‌های کاربران، هدایت آن‌ها به برنامه‌های مختلف و توزیع بار بین سرورهاست. همچنین به خاطر سرعت و کارایی بالا در مدیریت تعداد زیاد اتصال هم زمان شناخته می‌شه.

۲- وقتی یه کاربر یک آدرس وب را در مرورگر خود وارد می‌کنه، مرورگر ااول آدرس دامنه را به ای پی سرور تبدیل می‌کند و سپس یک اتصال شبکه با سرور برقرار میکنه. بعدش مرورگر یک درخواست برای دریافت صفحه یا داده به سرور ارسال میکنه. سرور این درخواست را دریافت میکنه، در صورت نیاز کدهای برنامه را اجرا می‌کنه و یا از دیتابیس اطلاعات میگیره. سپس نتیجه رو به صورت یک پاسخ به مرورگر میفرسته. در نهایت مرورگر این داده‌ها را پردازش کرده و صفحه‌ی وب را به شکل قابل مشاهده برای کاربر نمایش می‌دهد. در مرحله شکل گیری ارتباط یه TLS Handshake هم انجام میگیره.

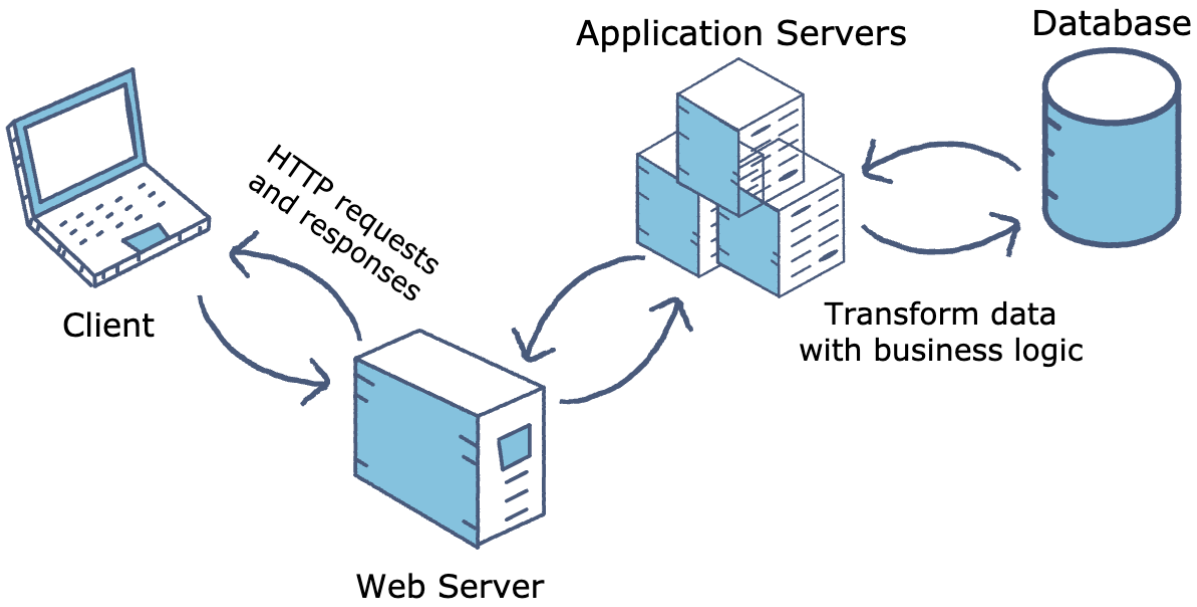
۳- HTTP پروتکلیه که مرورگر و سرور برای تبادل اطلاعات در وب از اون استفاده می‌کنن. وقتی یک URL را در مرورگر وارد میکنیم، ابتدا دامنه به آدرس IP سرور تبدیل می‌شود تا مرورگر بداند به کدوم سرور باید متصل بشه. بعدش مرورگر یک درخواست یا ریکوئست به سرور میفرسته تا صفحه یا داده مورد نظرو دریافت کنه. سرور این درخواست را پردازش میکنه، در صورت نیاز با دیتابیس یا برنامه‌های بک ارتباط می‌گیره و پاسخ یا همون ریسپان مناسب رو آماده می‌کنه. این پاسخ به مرورگر

ارسال می‌شه و مرورگر اونو پردازش کرده، منابع جانبی مثل CSS و java script را بارگذاری میکنه و در نهایت صفحه‌ی وب را به کاربر نمایش میده.

۴- اپلیکیشن Django در لایه (شاهکار)Backend قرار داره و وظیفه‌ی پردازش منطق برنامه و تولید محتوای داینامیک را بر أساس چیزی که دریافت میکنرو داره. وقتی کاربر در مرورگر یک URL را وارد می‌کنه یا هرکاری انجام میده، مرورگر یک درخواست HTTP/HTTPS به وب‌سرور(مثلNginx) می‌فرسته. وب‌سرور می‌تونه فایل های استاتیک مثل(تصاویر، CSSو JavaScript) را مستقیما ارسال کنه ولی برای درخواست‌های داینامیک، اونارو به اپلیکیشن Django ارسال می‌کنه .

Django درخواست را پردازش کرده و در صورت نیاز با دیتابیس یا سرویس‌های دیگر ارتباط برقرار می‌کنi. سپس پاسخ تولید شدرو به وب‌سرور برمی‌گردونه و وب‌سرور آن را به مرورگر می‌فرستد تا کاربر صفحه وب را مشاهده کنه.

۵-



وقتی کاربر یک URL را در مرورگر وارد میکنه، اول از همه DNS آدرس سرورو پیدا می‌کنه. بعدش مرورگر درخواستHTTP/HTTPS رو از طریق شبکه به وب‌سرور

ارسال می‌کنه. وب‌سرور درخواست را دریافت کرده و برای محتوای داینامیک اونو به اپلیکیشن Django می‌فرستد.

Django درخواست را پردازش میکنه و در صورت نیاز با دیتابیس ارتباط بقرار میکنه تا داده‌هارو بخونه یا ذخیره کنه. در نهایت پاسخ ساخته شده از طریق وب‌سرور به مرورگر برمی‌گرده و صفحه وب به کاربر نمایش داده می‌شه (عکسه ناقصه ولی قشنگ بود گذاشتم).

۶- Django یک فریم ورک وب سطح بالا برای پایتونه و هدفش اسان تر کردن توسعه وب و جداسازی بخش های مختلف برنامست.

معماری Django معمولا شامل سه بخش اصلی است که خودش دو مدل داره:

MVC (Model–View–Controller)

- Model: مدیریت داده‌ها و منطق برنامه را در دست داره
- View: رابط کاربری و نمایش داده‌ها به کاربران
- Controller: کنترل جریان برنامه، دریافت درخواست کاربرها و هدایت اون به Model و View هست.

دومین مدل:

MVT (Model–View–Template)

- Model: همون مدل در MVC
- View: درخواست ها را دریافت، منطق برنامه را اجرا و داده‌ها را آماده میکنه. عملا همون نقش Controller در MVC رو داره.
- Template: مثل View در MVC ، ولی فقط وظیفه نمایش داده‌ها به کاربر رو داره.

بخش دوم:

- **URL**

مسیر درخواست‌ها را مشخص و تعیین میکنه که کدوم View باید آن درخواست را پردازش کند. در جنگو، مسیرها در فایل urls.py تعریف میشن.

- **View**

درخواست کاربر را دریافت و پردازش میکنه. داده‌ها را از model میگیره، منطق برنامه را اجرا میکنه و پاسخ رو به Template می‌فرسته.

- **Model**

داده ها و منطق مرتبط با دیتابیس را مدیریت میکنه. مشخص میکنه که داده‌ها چگونه ذخیره شون و خوانده یا تغییر داده بشن.

- **Template**

مسئول نمایش داده‌ها به کاربرانه. داده‌های آماده شده توسط View را دریافت کرده و در قالب HTML به شکل قابل مشاهده و تغییرات گرافیکی نمایش میده.

به نحوی این مسیر طی میشه:

مرورگر → URL → View → (Model) → داده‌ها → View → Template → مرورگر

۷- یک محیط مجازی پایتون (virtualenv) و در واقع یک محیط ایزوله است که به هر پروژه اجازه میده که کتابخانه ها و بسته های مورد نیازشو جدا از سیستم اصلی نصب و مدیریت کنه. با استفاده از محیط مجازی، نسخه‌های مختلف پکیج ها بین پروژه های مختلف تداخل پیدا نمیکنن و هر پروژه میتونه نسخه مخصوص به خودشو داشته باشد. همنیطورم باعث میشه پروژه ها راحت تر روی سیستم‌های دیگه منتقل و اجرا بشن، چون تموم وابستگی ها و نسخه های مورد نیاز پروژه در همون محیط مشخص شدن. نکته اخرم اینکه در پروژه های جنگو یا هر پروژه پایتونی ای، استفاده از محیط مجازی به حفظ نظم،

ایزوله بودن، امنیت و اطمینان از اجرای درست برنامه کمک می‌کند و مدیریت کتابخونه هارو بهتر میکنه. و باعث نمیشه یه کدی داخل کامپیوتر شخصی کار بکنه ولی در لب تاب دیگه نه.

۸- Django Templates:

برای ساخت صفحات وب و نمایش داده‌ها به کاربران استفاده میشه. برای ساخت رابط کاربری و صفحات html که داده‌ها از view دریافت میشن و مناسب وب سایت هایسه که کاربر مستقیما با مرورگر تعامل داره.

Django REST Framework:

برای ایجاد API و ارسال داده‌ها به فرمت هایی مثل JSON یا XML استفاده میشه. این روش مناسب برنامه‌های موبایل، SPA یا سرویس‌های دیگره که نیاز به دریافت یا ارسال داده دارنه.