Task-1-web-architecture-and-django-foundations

General Web Concepts:

1.what is a web server?(for example, what is the role of nginx)

خب بیاییم وب سرور رو هم تو بخش نرم افزار و هم تو بخش سخت افزاری بررسی کنیم:

.1سخت افزار: خب میتونیم بگیم که میتونیم وب سرور رو یک کامپیوتری فرض کنیم که میاد ابجکت های سایت ما که شامل فایل های HTML,CSS,JavaScript و یا عکس یا ... رو همراه با اون بخش نرم افزاری که انجام میشه ذخیره میکنه. و در واقع میاد تبادل دیتا هارو با دستگاه هایی که به اون متصل میشن انجام میده.

.2نرم افزار: در سمت نرم افزاری وب سرور میاد نوعی دسترسی های کاربران مختلف رو تنظیم میکنه.مثلا یک سرور HTTP است). حالا خود سرور HTTP هم یک پروتکل از لایه اپلیکیشن هست که مخفف HyperText Transfer Protocol است که مشخص میکند Browser ما با وب سرور چگونه ارتباط برقرار کند و ساده تر بگویم اینکه وقتی که مثلا ما یک سایتی رو میبینیم مورد استفاده قرار میگیرد و دستور ها و قوانینی دارد که برای تبادل دیتاهایی که بالا گفتم استفاده میشود (و بیام حتی ساده تر هم توضیح بدم اینجوریه که وقتی ی سایت رو میبنیم browser ما به وب سرور request میفرستد و وقتی که دریافت میشود و تایید میشود یک responseارسال میشود و اینجوری میتوانیم ما از سایت دیدن کنیم.

حالا هر وب سروری یک IP Adrress و یک Domain name دارد.خب حالا برای اینکه خود کامپیوتر ما یک وب سرور بشود نیاز است که یک نرم افزار وب سروری رو مثل nginx رو نصب کنیم).میتوان از وب سرور های دیگه مثل Apache که مثل nginxمعروف هم هست استفاده کنیم یا(Tornado,XAMPP)

حالا بیام یه توضیح مختصری هم درباره nginx بدم. خب nginx علاوه بر وب سرور بودن چند تا کار بسیار مهم رو میتونه انجام بده .مثلا به عنوان reverse proxy که وظیفش اینه که یک واسطه ای بین client و server هست که درخواست هارو از کلاینت میگیره و به سرور میده و برعکس و مثلا یکی از مزایای این کار اینه که آدرس سرور مخفی بمونه. و مثلا یکی دیگه از وظیفش load balancingهست که مثلا اگر چند تا سرور داشته باشیم, کار هارو بین اونا تقسیم میکنه و میتونه به عنوان caching هم استفاده بشه.

به چند تا از مزایای nginx هم اشاره کنم:

.1پایین آوردن مدت زمانی که طول میشکه سایت بالا بیاد

.2بهتر کردن عملکرد با مسیریابی های بهتر

.3ارزان و قوی و مدیریت بهتر درخواست های موازی و همزمان

حالا معایبی هم دارد .مثلا سطح پشتیبانی آن پایین است و تو این مورد ضعیف تر از apache هست.

2.How do a client (browser) and server (backend) communicate?

خب تو توضیحات بالا درباره این سوال هم کمی توضیح دادم.ولی بیایین بهتر توضیح بدم.

همان طور که گفتم وقتی که یک سایت رو باز میکنیم مرورگر ما یک درخواست HTTP or HTTPS که شامل URL و هدر و .. میتونه باشه رو به سرور ما یا همون بک اند میفرسته و واز طریق پروتکل TCP/IPبه سرورمون وصل میشه و Request هارو از طریق TCP ارسال میکنه وسرور درخواست رو بررسی میکنه و نیاز کلاینت را در قالب Response ارسال میکنه که میتونه شامل عکس و HTML,CSS,JavaScript باشه و بعد اون مرورگر اون رو برای ما نمایش میده. حالا بعضی مواقع هم به جای درخواست یک صفحه وب میتوانیم دیتایی رو درخواست کنیم که اینجا میتوان از API استفاده کرد

3. What is HTTP? What happens when you visit a URL?

درباره HTTP هم یکمی صحبت کردم HTTP مخفف HyperText Transfer Protocol است که وقتی ما یک وب سایتی رو میبینیم مورد استفاده قرار میگیره HTTP.مجموعه ای قوانینی رو تعریف ست میکنه که برای ارتباط با وب سرور و ارسال دیتاهایی که چند بار گفتم, استفاده میشه.خب وقتی که ما از سایت بازدید میکنیم Browser ما request هایی رو برای ارسال به وب سرور و گرفتن دیتا ها درست میکنه.خب قبل این کار ما باید به Browser بگوییم که کجا و چه دیتا هایی رو باید بهمون بده .اینجاست که URL(Uniform Resource Locator)به کمک ما میاد URL.یک دستورالعملی هست که به ما میگه چجوری به منابع دسترسی پیدا کنیم و خودش هم از چند بخش تشکلیل شده که نیازی نیست الان توضیح بدمشون.(فقط اینکه شامل پروتکل و دامنه و مسیر و ... میشه)

حالا خود HTTP شامل متود ها و هدر ها و همچنین یک بدنه مشخص برای Request, Response ها داره. از متد های معروف اون میتونیم به GET اشاره کنیم که برای گرفتن دیتا از وب سرور استفاده میشود. همیچنین HTTP کد هایی دارد که یه status code معروف هستند .مثل 200 ok یا 404 not found که باهاشون آشنا هستیم.حالا هنگام بازدید از یکURL ، این از طریق DNS ، IP سرور رو به دست میاره و اتصال خودش رو برقرا میکنه.

4. Where does a Django app fit in the network architecture?

خب اینکه جنگو کجای شبکه قراره میگیره حالا ایشالله که سوال رو درست فهمیدم اینه که جنگو خودش یک وب فریم ورک سمت سرور هستش و با پایتون نوشته میشه و وقتی درخواست های HTTP فرستاده میشه , ابتدا توسط وب سرور جواب داده میشه و اون دیتا هایی که چند با هم اشاره کردم رو میفرسته و اگر هم درخواست های داینامیک مثل API رو داشته باشیم وب سرور اون رو به برنامه جنگو ما میفرسته و از اون به بعد بقیه کار ها بر عهده جنگو خواهد بود و Response رو بعدش میفرسته(ممکنه با دیتابیس تعامل داشته باشه و از این جور چیز ها).بعد خب فک کنم بازم تو سوال اول اشاره کردم که خود جنگو در لایه هفتم تو مدل OSI یا همان لایه اپلیکیشن قراره میگیره و سطحی که منطق برنامه هندل میشه و وظیفه ارسال هم بر عهده جمگو نیست و وب سرور و اتصالی TCP/IP که برقرار میکند این کار رو انجام میده.

5. Understand the following flow and describe what happens at each step:

Browser → DNS → Network → Web Server → Django App → Database

تقریبا در سوال های قبلی من این روند رو توضیح دادم. خب با همه استپ های اشنا شدیم به جز DNS(Domain Name Service) .اول DNS رو توضیح میدم و بعدش هر استپ رو توضیح میدم.خب برای ما قاعدتا ساخته که بیاییم به جای آدرس سایت (URL) بیاییم IP آدرس آن که) که مثلا فرمت IPv4 به این صورت هست 8.8.8.8 که هر کدام میتوانند بین 0-255 باشند (که خیلی سخت در یاد فرد میماند را حفظ کنیم. به زبان ساده تر همانند همه خونه ها که آدرس یونیکی در جامعه دارند , همه کامپیتور هایی که توی شبکه هستند هم دارای آدرس یونیک یا همان IP address هستند.خب اینجاست که DNS به کمک ما میاد تا ما به جای IP اسم اون سایت مثلا example.com رو حفظ کنیم.

خب همه این استپ هارو گام به گام توضیح میدم. وقتی که ما اول کار از سایت بازدید میکنیم به وسیله Browser خودمون ابتدا آدرس سایت مثلا همان example.com توسط DNS به IP سایت دسترسی پیدا میکنه و اون HTTP Request خودش رو از طریق ارتباط TCP/IP که تو شبکه برقرار کرده به وب سرور ارسال میکنه . وب سرور درخواست Browser را بررسی میکنه و اگر فایل استاتیکی درخواست شده باشه که خودش جواب درخواست رو میده و در غیر اینصورت اون رو به Django App ما پاس میده و Django Appهم با دیتابیس یا حتی سرویس ها و API های دیگه ارتباط برقرار میکنه و پاسخ رو ارسال میکنه.

Django and Backend Fundamentals:

6. What is Django’s architecture?

o Explain the concepts of MVC (Model–View–Controller) and MVT (Model–View–

Template) in the context of Django.

o You should clearly understand the roles of the following core components in

Django:

- URL
- View
- Model
- Template

خب بیاییم یه توضیح کلی بدیم و بعدش هر مورد رو تکی تکی توضیح بدیم.

میتونیم بگیم هر سایت از سه بخش اصلی input logic, business logic, and user interface logic: تشکیل شده است.

حالا با توجه به سرچ و تحقیق هایی که داشتم به این نتیجه رسیدم که MVC ,MVT دو معماری کاملا متفاوت نیستند و MVT نوعی پیاده سازی و تغییر یافته MVC هست و جنگو از MVT استفاده میکنه.

MVC (Model–View–Controller):

در این معماری

Model: مدیریت دیتا ها و ازتباط با دیتابیس و منطق بر عهده این بخش است

View: همان رابط کاربری یا UserInterface

Controller: رو مدیریت View, Model این قسمت هم همان طور که از اسمش معلومه وظیفه کنترل کردن درخواست ها بین میکنه

حالا تو جنگو تفاوت وجود دارد و از MVT استفاده میکند. بخش Model در هر دو معماری یکسانه.

در جنگو template وظیفه رابط کاربری رو داره و شامل HTML,CSS,JS هست.

بخش View منطق پردازش درخواست ها و تعامل رو انجام میده)مثل(Controller

حالا نقش های اصلی که تو صورت سوال گفته رو دقیق تر توضیح بدم:

1.URl:

این بخش همان طور که در سوال های قبلی درباره URL صحبت کردیم ، درخواست های HTTP رو به View منتقل میکنه

2.View:

این بخش هم داده هارو با توجه به درخواستی که آمده از بخش Model گرفته و تحویل Template میدهد.

3.Model:

در این بخش هم ساختمان برنامه ما تعریف میشود و تعامل با بقیه سرویس ها مثل دیتابیس در اختیار این بخش است.

4.Template:

این بخش هم وظیفه نمایش داده ها به کاربران رو برعهده داره و شامل فایل های HTML,CSS,JS است.

یعنی در آخر اینجوری بگم که وقتی یک URL وارد میشه,بخش URL میاد View مناسب رو صدا میزنه و View هم داده هم رو از Modelمیگیره و تحویل Template میده تا نمایش داده بشه.

7. What is a Python virtual environment (virtualenv)?

در واقع virtual environment یک محیط ایزوله هستش که تو اون میتوانیم ما پروژه های پایتون رو تست و ران بکنیم.

این به ما اجازه میده که بتوانیم پروژه خودمون رو به صورت مجزا از بقیه پروژه ها و حتی اون نسخه اصلی پایتون که روی سیستم ما نصب است, اجرا بکنیم.(دقیقا مثل یک کانتیر تو داکر).

تو virtual environment ما پکیج های مخصوصی که خودمان برای پروژه کنونی لازم داریم رو نصب کرده ایم و فقط تو این محیط هست و در خارج از این محیط خبری از اون پکیج ها نیست و همچنین این محیطی که درست کرده ایم از سایر محیط های virtual environmentجداست .حالا از مزایای اصلی این بگم اینکه از تداخل پکیج ها و ورژن های مختلف تو پروژه های مختلف جلوگیری میکنه و میتونیم بگیم خود نسخه اصلی پایتون رو که نصب کرده ایم رو تمیز تر نگه میداریم و یک مزیت اصلی همان طور که گفتم اینه که میتونیم ورژن های مختلف پایتون با پکیج های مختلف رو فقط تو خود همون محیط بالا بیاریم.

Why is it important in Django or any Python project?

حالا درباره این سوال هم توضیح دادم تو بخش مزیت های.virtual environment

مثلا یک پروژه ما نسبت به اون یکی پروژه نیازمند یک نسخه دیگری از جنگو است و در این صورت اگر هر دو روی یک محیط نصب شوند باعث تداخل میشه و برای حل این مشکل از virtual environment استفاده میکنیم.همینطور پکیج هایی که برای هر پروژه نیاز است نصب کنیم ,فقط بر روی همان محیط باقی میمانند و اگر هم کسی از ما پروژه رو بخواد میتونیم در داخل یک فایل requirementهای پروژه را اعلام کنیم.

8. What is the difference between Django REST Framework and Django Templates?

o Compare their purpose, usage, and when to use each one.

خیلی ساده بخوام بگم Django Templates خودش فرانت رو هم هندل میکنه و خروجی این شاملHTML، CSSو JS است و در حالی که DRF خروجیش داده های متنی در قالب Json یا سایر قالب ها هست و برای ایجاد API ها برای ارسال و دریافت داده ها استفاده میشود.مقلا وقتی که میخواییم خودمون سایتی داشته باشیم که فرانت هم داشته باشه و تو Browser بتونیم..., HTML رو ببینیم از Templates استفاده میکنیم , ولی اگر ما فقط میخواییم بک رو بزنیم و فرانت هم مستقل از بک هستش از REST استفاده میکنیم.حالا درباره مورد استفاده هر کدوم هم همانطور که گفتم وقتی میخواییم وب سایتی داشته باشیم که با صفحه های کامل و اینا نیازمند استفاده از Django Templates هستیم و در غیر اینصورت اگر بخواییم API محور کار کنیم و جدا از فرانت از Django REST Frameworkاستفاده میکنیم.در واقع هدف ما در Django REST Framework ارائه داده به سایر برنامه ها هست.

Hands-On Setup Questions:

9. How can we create a simple Django project?

o Mention the basic commands and folder structure

خب اول کار باید بیاییم یک Venv رو بسازیم.

سپس اون رو فعال کنیم و بعد از اون حالا من چون خودم جنگو رو سیستم نداشتم و باید جنگو رو نصب میکردم). اگر هم داشتم هم دوباره باید باpip نصب میکردم چون داخل Venv هستیم(

بعد از اون یک پروژه جنگو ساختم و وارد پروژه شدم و اون رو ران کردم

همه دستورات مرحله به مرحله به این صورت هستند:

```
python3 -m venv task1
```

```
source task1/bin/activate
```

```
pip install django
```

```
django-admin startproject TaskProject
```

```
tree TaskProject/
```

```
TaskProject/
├── manage.py
└── TaskProject
    ├── asgi.py
    ├── __init__.py
    ├── settings.py
    ├── urls.py
    └── wsgi.py
```

2 directories, 6 files

cd TaskProject/

python manage.py runserver

10. How can we create an app inside a Django project?

o Explain how apps are organized and how they are connected to the main

project.

خب app میتونیم بگیم در داخل جنگو یه واحد مستقل هست که میتونه view ها و templateهای خودش رو داشته باشه و پروژخ اصلی فقط اون هارو مدیریت میکنه و مسیر های آن رو به کل سایت متصل میکنه.

حالا برای این کاری که سوال خواسته ما به مسیری میرویم که manage.py در آن قرار دارد.

در آن یک app ایجاد میکنیم و حالا باید اون رو به پروژه اصلی متصل بکنیم و سپس url ان را تنظیم کنیم.

python manage.py startapp app_1

app_1/

├── admin.py

├── apps.py

├── __init__.py

├── migrations

| ├── __init__.py

| └── __pycache__

| └── __init__.cpython-312.pyc

├── models.py

├── __pycache__

```
|   ├── admin.cpython-312.pyc
|   ├── apps.cpython-312.pyc
|   ├── __init__.cpython-312.pyc
|   ├── models.cpython-312.pyc
|   ├── urls.cpython-312.pyc
|   └── views.cpython-312.pyc
├── tests.py
├── urls.py
└── views.py

4 directories, 15 files
```

in setting.py:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app_1',
]
```

and add path('app_1/', include('app_1.urls')), to TaskProject/urls.py

Write a short explanation of what happens when visiting a URL (answering the questions), including:

o network

o Request path

o URL dispatcher

o View handling

o Template rendering (if used)

خب برای جواب دادن به این سوال با توجه به توضیحاتی که در بخش های قبلی داشتیم:

o network:

در ابتدا Browser ما یک درخواست HTTP به سرور که همان لوکال هستش میفرسته و سرور این request رو گرفته و تحویل جنگو میده.

o Request path:

حالا برنامه ما با توجه path درخواست تشخیص میده که کدام app استفاده بشه

o URL dispatcher:

اگر با یک url مطابق بود , درخواست آن رو به view مربوط ارسال میکنه.

o View handling:

حالا با توجه به منطق برنامه view خودش منطق برنامه را اجرا میکنه

o Template rendering (if used):

که حالا اگر از قالب های HTML,... استفاده کرده بودیم که نکردیم , دیتا ها به صورت آن قالب ها به browser ارسال میشدند.