

Task 1: Web Architecture and Django Foundations

Objective:

This task is intended to give you a solid understanding of core web development concepts, including how Django fits into the web architecture. It is one of the most important tasks in the course, and you are expected to research and answer all of the following questions in detail.

Questions to Explore:

General Web Concepts

1. What is a web server? (for example, what is the role of nginx)
2. How do a client (browser) and server (backend) communicate?
3. What is HTTP? What happens when you visit a URL?
4. Where does a Django app fit in the network architecture?
5. Understand the following flow and describe what happens at each step:
Browser → DNS → Network → Web Server → Django App → Database

Answer:

وب سرور یک برنامه کامپیوتری است که درخواست های کاربران را از طریق مرورگر یا سایر کلاینت های **HTTP** دریافت میکند و محتوای وب را به مرورگر یا کلاینت ارسال میکند **NGINX** یک وب سرور متن باز است که به دلیل عملکرد بالا و مصرف کم منابع شناخته میشود این وب سرور در ارائه محتوای استاتیک بسیار کارآمد بوده و همچنین با سیستم عامل های مختلف بسیار سازگار است

ارتباط بین کلاینت و سرور عمدتاً از طریق **http** انجام میشود

http یک پروتکل ارتباطی بین سرور و کلاینت میباشد این پروتکل شامل مجموعه ای از قوانین است که نحوه ارتباط با مرورگر های وب و سرور ها را برای انتقال داده ها تعیین میکند . زمانی که از یک **url** بازدید میکنیم مرورگر از **http** برای ارسال درخواست استفاده میکند و آن را به سرور میفرستد .

برنامه جنگو در لایه برنامه ی معماری شبکه قرار میگیرد برنامه جنگو منطق برنامه وب را مدیریت میکند و درخواست ها را پردازش میکند و ارتباط با دیتابیس را فراهم میکند زمانی که یک **url** را در مرورگر وارد میکنم مرورگر با استفاده **dns** ادرس (**ip**) را بدست می آورد و

یک درخواست در قالب **http** به سرور ارسال میکند وب سرور آن را دریافت میکند و آن را بررسی میکند اگر یک فایل استاتیک را بخواهد آن را مستقیماً ارسال میکند اگر آن نیاز به پردازش داشته باشد آن را به بک اند میفرستد برنامه بک اند آن را پردازش میکند و با دیتا بیس ارتباط برقرار کرده و در نهایت نتیجه درخواست را به وب سرور میفرستد و وب سرور آن را به مرورگر جهت نمایش به کاربر ارسال میکند

Django and Backend Fundamentals

6. What is Django's architecture?

- Explain the concepts of MVC (Model–View–Controller) and MVT (Model–View–Template) in the context of Django. ○ You should clearly understand the roles of the following core components in Django:
 - ✦ URL
 - ✦ View
 - ✦ Model
 - ✦ Template

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

- Why is it important in Django or any Python project?

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

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

Answer:

جنگو از معماری **MVT** استفاده میکند
معماری **MVT** شامل :
mode مسئول مدیریت داده ها و ارتباط با دیتا بیس را دارد
view نقش **contoroller** را در معماری **mvc** دارد
template مسئولیت نمایش اطلاعات را به کاربر دارد

معماری MVC :

mode: همانند معماری **mvt** میباشد

view : مسئولیت نمایش اطلاعات را به کاربر دارد

contoroller : واسط بین **view** و **model** که منطق برنامه را مدیریت میکند

URL : تعیین میکند هر ادرس به کدام **view** متصل شود

view : درخواست را پردازش میکند و پاسخ مناسب میدهد

model : وظیفه ارتباط با دیتابیس را دارد و مدیریت ساختار داده ها

template : مسئولیت نمایش اطلاعات را به کاربر دارد

Python virtual environment : یک محیط مجازی ای ایزوله را برای پروژه فراهم میکند

تا برنامه نویس بتواند پکیج های مورد نیاز را مستقل از پکیج های موجود بر سیستم به

پروژه اضافه کند و از تداخل های ناخواسته نیز جلوگیری شود

Django Templates : داده ها در قالب **html,css,...** ارسال میکند و هنگامی که بخواهیم

یک صفحه را کامل بفرستیم از ان استفاده میکنیم (فقط میتوانیم برای مرورگر یا

برنامه های تحت وب بفرستیم)

Django REST Framework : داده ها را به صورت **json,xml** ارسال میکند و هنگامی که

برنامه فرانت جدا باشد یا کلاینت های مختلفی داشته باشیم استفاده میشود

Hands-On Setup Questions

9. How can we create a simple Django project? ○ Mention the basic commands and folder structure.

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

- Explain how apps are organized and how they are connected to the main project.

Answer:

ابتدا جنگو را با دستور **pip install Django** نصب میکنیم سپس یک فولدر ساخته و وارد آن میشویم و برای ایجاد یک پروژه جنگو دستور **Django-admin startproject <name>** در ترمینال وارد میکنیم برای ساخت **app** دستور **python manage.py startapp <name>** وارد میکنیم سپس در قسمت **settings.py** در بخش **INSTALLED_APPS** اسم **app** را اضافه میکنیم تا به پروژه اصلی متصل شود

Deliverables:

To complete this task, you must:

- **Set up Python and Django in a virtual environment**
- **Create a Django project and an app**
- **Build a simple Django app that returns: Hello World <your name> in the browser**
- **Write a short explanation of what happens when visiting a URL (answering the questions), including:**
 - network
 - Request path
 - URL dispatcher
 - View handling
 - Template rendering (if used)

Answer : زمانی که از یک **url** را وارد میکنم مرورگر یک

درخواست **http** را به سرور جنگو ارسال میکند سرور آن را

به جنگو میفرستد جنگو آن **url** را با **path** های موجود

مقایسه و در صورت تطابق آن را به **view** مرتبط ارسال

میکند تا در **view** بررسی و جواب مناسب به کلاینت داده

شود گاهی اوقات در خواست در یک فرمت مشخص در

قالب یک کد **html** رندر میکند (template)

Submit this explanation as a PDF or Markdown (.md) file

Push your entire project (including the code and explanation document) to the Git repository.

Sources:

What happened when we see a web page:

<https://www.youtube.com/watch?v=0oLPmDgEqTI>

What is http:

<https://www.youtube.com/watch?v=2yfDgnm6eAs>

Role of nginx:

<https://devoriales.com/post/339/nginx-101-a-comprehensive-tutorial-for-developerspart-1-2> <https://www.youtube.com/watch?v=iInUBOVeBCc&t=10s>

What is Django:

<https://www.interviewbit.com/blog/django-architecture/>

https://www.youtube.com/watch?v=t_p4ZyAYyaY

Start with Django:

<https://www.geeksforgeeks.org/python/python-web-development-django/>

<https://www.youtube.com/watch?v=nGlg40xs9e4&t=103s>

Python learning:

Python concepts we should know: basic python i/o handling, data structures, object-oriented programming concepts, debugging and etc ...

Mosh crash course:

<https://www.youtube.com/watch?v=uQrJ0TkZlc>

Book: head first python

Use websites like <https://leetcode.com/> to get your hands dirty. (it is very important)

Complete Course : <https://www.udemy.com/course/complete-python-bootcamp/> (not recommended to see all the videos but can be downloaded from downloadly)