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

---

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.**

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.**

✅ 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)

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-developers-part-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=nGIg40xs9e4&t=103s

Python learning:

Python concepts we should know: basic python i/o handling, data structures, object-oriented programing 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)