Exercise 2.1: Getting Started with Django

Learning Goals

- Explain MVT architecture and compare it with MVC
- Summarize Django's benefits and drawbacks
- Install and get started with Django

Reflection Questions

1. **Suppose you're a web developer in a company and need to decide if you'll use vanilla (plain) Python for a project, or a framework like Django instead. What are the advantages and drawbacks of each?**

    Advantages of Vanilla Python:
    - Flexibility: You have complete control over the structure and design of your application.
    - Lightweight: No need to load a large framework, which can be beneficial for smaller or simpler projects.
    - Learning Opportunity: Helps in understanding the core principles of web development without any abstraction.

    Drawbacks of Vanilla Python:
    - Time-Consuming: Developing common functionalities (like authentication, routing, form handling) from scratch can be time intensive.
    - Maintenance: More effort required to maintain and scale the application.
    - Security: Increased risk of vulnerabilities if not handled properly, as you have to manage everything manually.

    Advantages of Django:
    - Batteries-Included: Comes with a lot of built-in features such as authentication, ORM, admin interface, and more, which speeds up development.
    - Scalability: Designed to handle high traffic and can scale well with your application.
    - Community and Support: Large community and extensive documentation make it easier to find help and resources.
    - Security: Includes built-in protections against common security threats like SQL injection, CSRF, and XSS.

    Drawbacks of Django:
    - Learning Curve: Can be complex to learn, especially for beginners.
    - Overhead: Might be overkill for small projects due to its size and the number of features it includes.

- Abstraction: Can hide the underlying processes, which might limit understanding of core web development concepts.

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?

   The advantage of MVT architecture over MVC architecture is the simplified handling of the view component. In MVT, the view is more focused on data presentation and is directly connected to templates, making the code cleaner and easier to manage. This integration between views and templates reduces boilerplate code and improves the efficiency of rendering dynamic content, leading to faster development and easier maintenance.

3. Now that you've had an introduction to the Django framework, write down three goals you have for yourself and your learning process during this Achievement. You can reflect on the following questions if it helps:

   - What do you want to learn about Django?
     - Learn about Django's features such as ORM, middleware, and forms.
     - Understand best practices for structuring Django projects and applications.

   - What do you want to get out of this Achievement?
     - Apply the knowledge gained to contribute to Django-based projects in a professional setting.

   - Where or what do you see yourself working on after you complete this Achievement?
     - After completing this Achievement, I see myself working on more complex web applications, possibly in a team setting where Django's features can be used to streamline development. I also aim to contribute to open-source Django projects to further improve my skills and give back to the community.