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?
    a. Advantages:
        i. Fast Development:
            - Django's MVT architecture ensures development is fast and easy.
        ii. DRY principles:
            - Django follows the DRY (Don't Repeat Yourself) principles, which helps to keep your code non-repetitive, non-redundant, and efficient.
        iii. Scalability:
            - Due to MVT, Django is a loosely coupled architecture, which makes it simple to add new hardware and components, which makes scalability easier.
    b. Disadvantages:
        i. Highly Structured:
            - Django likes to do things a certain way as it is a highly structured framework. This makes it difficult if a developer wants to have a lot of control over finer details and how the system works.
        ii. Not always suitable for small projects:
            - With a lot of its out-of-the-box solutions, its strengths are better suited for projects that need these solutions. Django's prewritten code means it's more server intensive and makes it rather heavy for low-bandwidth systems.

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture?
    a. One of the primary differences is that MVC requires a separate controller layer that developers have to manually implement, which can make for more complex and repetitive code. MVT on the other hand, handles the controller logic behind the scenes, which helps to reduce the amount of code developers need to write and maintain.

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?
        - I'm curious to learn what Django is capable of and how best to use it. What types of projects would I want to try and work on in the future with it and how I'd go about building them.
    - What do you want to get out of this Achievement?
        - I mostly just want to become more proficient in using the Django framework. If I can walk away feeling like I took a good amount of knowledge away and feel comfortable starting my own projects, I'll be pretty happy.
    - Where or what do you see yourself working on after you complete this Achievement?
        - After looking up some companies that use Django, I think it'd be fun to try and make something like a small music player app. I'd think it'd be interesting to learn more about how an application like that works.