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?

> **Vanilla Python** offers full control and flexibility since you write everything from scratch, which can be useful for very simple or highly customized projects. However, it requires more development time because you have to build many features yourself, like routing, authentication, and database handling.
> **Django**, on the other hand, provides a ready-made, structured framework with many built-in tools (authentication, admin panel, ORM) that speed up development and help maintain clean, scalable code. The drawback is that it enforces certain conventions and project structures, which might feel restrictive if you want complete control or are building very simple applications.

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

The biggest advantage of MVT over MVC is that MVT removes the need for developers to write the controller logic themselves by shifting that responsibility to the framework's template system. This means developers can focus more on defining what data to show (the View) and how to present it (the Template), while Django handles the control flow automatically, simplifying development.

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.

> I want to **understand the MVT architecture deeply**, including how Models, Views, and Templates interact within Django projects.
> I aim to **build a fully functional Django app** from scratch to gain hands-on experience with routing, database models, and user authentication.
> I hope to **learn best practices for scalable and secure Django development**, preparing me for real-world projects or professional roles involving Django.