

Exercise 2.1: Getting 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?

Choosing between plain Python and Django for your project is a big call. With plain Python, you've got total freedom—you can build everything exactly how you want it, which is perfect for smaller or highly customized projects. Django, on the flip side, is like a multi-tool for web development. It's loaded with features right out of the gate, which can save you a ton of time. It's ideal if you need to get a complex web app up and running quickly.

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

When it comes to MVT vs. MVC, here's why I prefer MVT: it simplifies the whole controller aspect. In MVT, Django's URL dispatcher and views work together to take care of what would normally be the controller's job in MVC. This makes the process smoother and more tailored to Django.

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?
 - What do you want to get out of this Achievement?
 - Where or what do you see yourself working on after you complete this Achievement?
-
- **Get comfortable with Django's ORM. Being able to interact with databases without writing raw SQL sounds like a game-changer.**
 - **Dive into Django's built-in admin interface. It seems like it could save a ton of time when managing the backend.**
 - **Build a full-fledged web app from scratch. Nothing beats hands-on experience, right?**