

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

- 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?

I would use a framework like Django because it makes web development so much easier. Django comes with built-in tools for handling databases, security, and user authentication, so I wouldn't have to code everything from scratch.

If I used vanilla Python, I'd have to manually write code for everything, like how data is stored, how users interact with the site, and even security features. That would take a lot more time and effort. So unless I had a very specific reason for avoiding Django, I would definitely use it.

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

When using MVT architecture, you don't have to write code to get the data from the database and map it to the URL like you would when using MVC.

All you would need to do is define the items to display to the user and the framework so that the template can prepare and send it to the user.

- 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 want to understand how Django actually works behind the scenes. Right now, it feels like a big system with a lot of moving

parts, and I want to learn what each part does. I know it makes web development easier, but I want to dig deeper into how it organizes data and manages websites.

- What do you want to get out of this Achievement?

I want to build a fully functional recipe website where users can add, search, update, and delete recipes in a way that looks good and is easy to use.

Right now, my app is just in the command line stage, but I want to bring it to life with a real user interface and to create an engaging experience for the users.

- Where or what do you see yourself working on after you complete this Achievement?

Once I finish this project, I'll start working on my own recipe app project that has even more features. I'm not sure yet if Django already includes all the tools I need, but whatever I learn here will help me build a strong foundation for my project.

Also, I'll be focusing on finding a job in web development. Learning Django is a big step toward that, and I want to feel confident using it in real-world projects.