Exercise 2.1: Getting Started with Django

Learning Goals

- Explain MVT architecture and compare it with MVC

In Model-View-Template pattern, Model, similar to MVC handles the data and logic part, maintains a connection with the database.

View works differently than the View-layer of MVC. Here, the view accepts the requests and sends out the response.

Template, this is the presentation layer. It is commonly coded in HTML or DTL (Django Template Language).

MVT Pattern is used by Django. So, when a request comes, Django runs it through different URL patterns to match it with the request. Then, the request is rendered to the particular view which then coordinates with Model for Data and Template for Presentation to send back a processed response.he user interface. They handle the presentation logic and rendering of data. The Template receives data from the Controller (through the View) and determines how it should be displayed. This separation of concerns allows for a clear distinction between the structure of the UI and the data it displays.

- Summarize Django's benefits and drawbacks

Benefits

Django Is Implemented in Python which is easy to read, has a lot of powerful computational features, and offers support in the backend without compromising features at the frontend.

Django's MVT architecture ensures development is fast and easy.

Thanks to its MVT architecture, Django handles network transmission and content delivery at a high speed.

Django Follows DRY (Don't Repeat Yourself) Principles.

Django provides an admin interface that allows easy set up of CDNs and content management.

Scaling a Django-based system isn't a problem.

Django is built with security in mind, allowing secure-by-design implementation with built-in automated encryption.

Django is open source and has a huge community of contributors. As such, it's usually very easy to get support if you need it.

Drawbacks

Django is a highly structured web framework that does things a certain way and there's no wiggle room to diverge from the rules.

Django Isn't Always Suited for Simple Projects that don't require database access or file management.

- 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? Vanilla Python will be much more lightweight and allow for more flexibility without Django's strict rules. Otherwise, developing with Django would be faster, more secure, and scale easier.

2. In your own words, what is the most significant advantage of Model View Template (MVT) architecture over Model View Controller (MVC) architecture? I still don't understand this but the exercise says: With MVT, you don't have to write code to fetch the data from the database and map it to the URL. You simply need to specify which items to present to the user, and the framework (Template) will prepare and send it.

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 want to understand why MVT is better than MVC
   - What do you want to get out of this Achievement? To understand Django enough to implement it in a project.
   - Where or what do you see yourself working on after you complete this Achievement? I will likely implement django for the backend of a react frontend app that I have been planning.