

Exercise 2.4: Django Views and Templates

Learning Goals

- Summarize the process of creating views, templates, and URLs
- Explain how the “V” and “T” parts of MVT architecture work
- Create a frontend page for your web application

Reflection Questions

1. Do some research on Django views. In your own words, use an example to explain how Django views work.

Django views are functions or classes written in Python that receive a web request and return a web response. The response can be a simple HTTP response, an HTML template response, or an HTTP redirect response. Developers can use these views to create logic to process and generate data, making it easier to build more complex apps.

2. Imagine you're working on a Django web development project, and you anticipate that you'll have to reuse lots of code in various parts of the project. In this scenario, will you use Django function-based views or class-based views, and why?

I would use class-based views in this scenario. Due to their class based nature, CBVs are easy to reuse and extend. They reduce the effort spent duplicating code, ultimately saving time. CBVs would be perfect for this scenario.

3. Read Django's documentation on the Django template language and make some notes on its basics.
 - Templates are stored in separate folders with .html extensions
 - Templates contain variables that are replaced with values when the template is evaluated.
 - Templates use tags to control the logic of the template.
 - Templates allow you to modify variables using filters.
 - Template files contain both static and dynamic content