

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 part of a MVT architecture, they are responsible for handling the logic of application, processing the user requests, interacting with models and returning appropriate responses. They are either function based, or class based.

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?

Class base view, they are based on python classes, and they are easy to reuse. Due to their class-based nature they reduce the effort spent duplicating or rewriting code.

3. Read Django's documentation on the Django template language and make some notes on its basics.

Templates: A text file that can generate any text-based format.

Variables: Gets replaced with values when the template is evaluated.

Tags: Control the logic of template. {%tags%}. More complex than the variables. Some create text in the output, some control flow by performing loops or logic, and some load external information into the template to be used by later variables.

Filters: You can modify variables for display by filters.

Comments: To comment-out part of a line in a template, use the comment syntax: `{# #}`.

Template Inheritance: Template inheritance allows you to build a base “skeleton” template that contains all the common elements of your site and defines **blocks** that child templates can override.