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.

The way Django views work is by handling logic under the rendering templates, which processes the submissions and performs other actions which are based on the user's request. The views section retrieves from the models, which then passes it to the templates and then generates the HTML content that is being sent to the users interface. When you define the views and associate them with URLS, Django is able to provide a clean and organized way to be able to handle different actions and showcase the dynamic web page.

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?

If I anticipate that I would need to reuse lots of code in various parts of the project I would use Django class based views instead of function based views. The class based views is able to give you a modular and reusable approach which allows for code organization and reduces code duplication within your project. Being able to have functionality, override and inherit code s=using class-based views increase development and code quality.

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

The Django template is a powerful language that comes with Django web framework. Templates provides a syntax to make dynamic web pages which combines HTML markup with template tags, variables and filters.