Challis Regan

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.
   1. A Django view function (or just “view”) is a Python function that takes a web request and returns a web response. The web response can be an HTML web page, JSON data, almost anything. The view is written in the view.py file and it is connected to a template. For example, YouTube.com is built with Django. The different URL destinations lead to different views and templates. There is the home page view, the view of an individual video, the view of a YouTube creator’s channel, and your own channel’s view.
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?
   1. I would use class-based views because CBVs are optimized for repeating code. For example, I could make one function and use it in several apps. I would not use function-based views because it would be harder to re-use or extend code.
3. Read Django’s documentation on the [Django template language](https://docs.djangoproject.com/en/3.2/ref/templates/language/#templates) and make some notes on its basics.
   1. A template is a text file which can generate any text-based format (HTML, XML, CSV, etc.). A template contains variables. Variables get replaced with values when the template is evaluated. A template also contains tags, which control the overall logic. Variables are written with brackets, like this: {{ variable }}. When the template engine encounters a variable, the variable is evaluated and replaced with the result. Tags are like variables, but more complex. Tags are written with brackets and percent signs: {% tag %}. Django has about two dozen built-in template tags that do things like create text in the output, control flow with loops and logic, load external information into the template, etc.

Sources:

<https://docs.djangoproject.com/en/5.1/topics/http/views/>

<https://djangostars.com/blog/10-popular-sites-made-on-django/>

<https://docs.djangoproject.com/en/5.1/ref/templates/language/>