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

- Do some research on Django views. In your own words, use an example to explain how Django views work.
    - Django views handle what a web page shows when someone visits it. A view can be a function or a class, and it takes a request (like clicking a link) and returns a response (like showing a webpage). Views are stored in a file called views.py inside an app’s folder.
- 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 (CBVs) when I need to reuse code across multiple pages because they help keep things organized and reduce repeated code.
    - For small projects, function-based views (FBVs) might be simpler, but for larger projects with similar views, CBVs make it easier to manage.
    - For example, if I have several pages that display lists of recipes, a ListView (a type of CBV) makes this easier than writing the same logic repeatedly.
- Read Django’s documentation on the Django template language and make some notes on its basics.
    - A template is a file that helps generate web pages using HTML, text, or other formats.
    - Templates are useful for keeping HTML separate from Python logic, making it easier to manage a website’s design.
    - Django’s templates are flexible and can be used for different file formats, including emails and CSV files.