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 an essential part of the Django framework, acting as the bridge between your models (the data layer) and your templates (the presentation layer). Views handle the logic required to process a user’s request, interact with the model, and then return the appropriate response (often an HTML page) to the user.

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?
   In this situation you would use class-based views. This would be the case because class-based provides generic views that can be used and the generic views can be extended and customized so developer can use reuse code.

3. Read Django’s documentation on the Django template language and make some notes on its basics.
   The template system uses dot-lookup syntax to access variable attributes. In the example of {{ question.question_text }}, first Django does a dictionary lookup on the object question. Failing that, it tries an attribute lookup – which works, in this case. If attribute lookup had failed, it would’ve tried a list-index lookup.