## 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 work in where a function or a class receives a web request and then returns a response. First a view is called. Then the views are passed to an HttpRequest object when called from the web application in the browser. Finally, the views generate a response.

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 functionbased views or class-based views, and why?

I would use class-based-views in this scenario. CBVs are a better option when in need of a generic view that must be reused across several apps as they have built-in views and can avoid duplication. Due to their class-based nature, CBVs 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.

Django's template language is designed to strike a balance between power and ease. It's designed to provide a bridge and make those who work with HTML, CSS, JavaScript, etc. feel comfortable. It is specifically built for developers to embed Django logic codes into HTML template files.