Exercise 2.4: Django Views and Templates

Reflection Questions :

- *Do some research on Django views. In your own words, use an example to explain how Django views work.*

  - **A:** Django views are Python functions and/or classes that take web requests and return web responses. They dictate what content should be displayed when a particular URL is accessed. A good beginners' example is what I worked on this this Task (2.4). The exercise and task guided me to set up what would be rendered as the "welcome page". This is what will be rendered when there is no additional endpoint on the URL, and is determined by the view.py function in the app.

- *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?*

  - **A:** Django's class-based views provide a more object-oriented approach, allowing behavior to be encapsulated in classes. This makes the code more reusable, especially when using the built-in generic class-based views, which also reduces boilerplate code.

- Read Django's documentation on the Django template language and make some notes on its basics.

  - **A:**
    - Unlike classes in past languages I've worked with, Django uses {curly braces} for template tags, and {{double curly braces for substitutions}}
    - Filters can be applied to variables with the ' | ' pipe character.
    - Tags are enclosed in {% curly braces with percent signs %}
      - e.g. {% if %} , {% for %} , {% block %} , {% extends %}

- Comments are written with {# curly braces with pound signs #}
- Parent templates can extend their structure to child templates
- Django always looks for templates in the ‘templates’ directories of each app.