

## Exercise 2.5: Django MVT Revisited

### Learning Goals

- Add images to the model and display them on the frontend of your application
- Create complex views with access to the model
- Display records with views and templates

### Reflection Questions

1. In your own words, explain Django static files and how Django handles them.

A static file in Django is anything that holds information that does not change on a website. Some good examples of static files are background images or place holder images that are initially provided by the developer. After defining the `STATIC_URL` in `settings.py` and storing files in a 'static' folder, `{% load static %}` can be used in templates to access the files.

2. Look up the following two Django packages on Django's official documentation and/or other trusted sources. Write a brief description of each.

Package	Description
ListView	Django class-based view. Renders multiple instances of a table in order to display a list objects.
DetailView	Django class-based view. Allows for the full details of a single object to be displayed.

3. You're now more than halfway through Achievement 2! Take a moment to reflect on your learning in the course so far. How is it going? What's something you're proud of so far? Is there something you're struggling with? What do you need more practice with? You can use these notes to guide your next mentor call.

Overall, I find myself really enjoying working in Django and I love the way it pairs with Python. I can see how developers with other habits could be put off by "the Django way" but I find it to be helpful to have a structure within which to work. I am proud of the way I am picking up on the flow of building Models, Views, and Templates. However, I definitely think I need more practice when it comes to all of the steps involved in registering and mapping URLs, and writing comprehensive tests.