

## 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.

Django static files are files used by the application that do not change dynamically – they can include images that are used for specific purposes, CSS, and JavaScript. They are files that the developer provides during the creation of the project and are stored in a static folder – so, unlike user-generated files (e.g. images they upload), the database doesn't need maintain a record of these static files as long as Django knows where to look for them.

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	When the ListView is executed, the view will display a list of objects that is retrieved via <code>self.object_list</code> . Records are stored in this <code>object_list</code> when the view is executed and can be listed by looping through the list.
DetailView	When the DetailView is executed, the view will operate on whichever instance is found via <code>self.object</code> . It is used to display detailed information for the specific object.

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.

So far learning how to use the Django framework has been a lot to take in but I've been enjoying it. It feels good to be back into making a web application versus a command line one, since I can see the changes I'm making and the visual really helps keep me motivated. I've also enjoyed the structure of the framework in that everything has its place and formatting. I know the exercises mentioned that a lot of developers feel stifled by the 'Django way' but I'm someone who thrives with a set structure.