

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. Django static files are usually added during the development process by the developer and are assets that make up the application, such as static images, JS files, CSS files, etc.
 - b. In Django, static files have to be stored in a specific folder under each app. In the respective template, `{% load static %}` must be called before any content is generated and another specific command to fetch the image from the static folder in the place where it should be loaded.
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	A page representing a list of objects. Django class-based view that simplifies displaying a list of objects from a database. It automatically handles querying, pagination, and rendering to a template, making it ideal for presenting collections like articles or products.
DetailView	Django class-based view for displaying a single object from a database. It retrieves an instance based on its primary key or slug and renders it in a specified template, perfect for showing detailed information about items like blog posts or products.

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.

- a. I am getting quite tired, so the last meters of the development marathon are going to be tough, despite Django and Python are exciting topics and I like to see how the app grows.
- b. I still don't fully get the Django way of programming, all the interconnections between views, templates, creating urls and registering them, etc. So I'll need more practice.