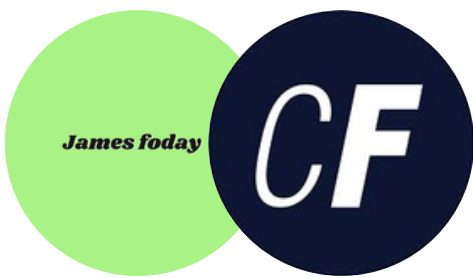James foday

CF

REFLECTION
QUESTION

Reflection Questions Answers

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

Django static files are files such as CSS, JavaScript, and images that are served directly to the client's browser to style and enhance the frontend of a web application. Django manages static files using the STATICFILES_DIRS and STATIC_ROOT settings. During development, static files are served automatically by Django's development server. For production, static files are collected into a single directory using the collectstatic command, which is then served by a web server such as Nginx or Apache. This approach optimizes delivery, caching, and organization of static resources.

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 generic class-based view in Django used to display a list of objects from a queryset. It simplifies building pages that show collections of database records, providing built-in pagination and filtering options.
DetailView	A generic class-based view in Django designed to display detailed information about a single object retrieved via its primary key or slug. It handles fetching the object and rendering it in a template automatically.

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.

I have learned a lot about Django's Model-View-Template (MVT) architecture and how to connect backend data with frontend views effectively. I'm proud of implementing image uploads and dynamic linking between list and detail pages with Django's class-based views. However, I am still working on mastering advanced URL routing and testing best practices. I want to practice more on writing robust automated tests and handling complex relationships between models.