

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

Django views are the “middleman” between my app models and templates. They control what data gets displayed to the user and what templates are used to present that data. When a user makes a request, the view decides how to respond to that request.

2. 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?

If I were working on a project that requires a lot of reusable code, I would prefer class-based views (CBVs). The reason is that CBVs provide a more organized and efficient way to manage repeated logic. They allow me to easily inherit and extend existing views, making it easier to share and reuse code.

3. Read Django’s documentation on the Django template language and make some notes on its basics.

The Django Template Language (DTL) is a way to dynamically generate HTML by embedding Python-like code inside my templates. It allows me to separate the presentation layer from the logic, making my web pages cleaner and easier to manage. Like template inheritance allows me to define a base template with common structure (like headers and footers) and extend it in other templates. This makes it easier to manage layouts.