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

In Django, views are Python functions or classes that handle HTTP requests and return HTTP responses. They act as the controller in the MVT (Model–View–Template) pattern — retrieving data from models, passing it to templates, and returning the rendered HTML to the browser.

Example:

python

CopyEdit

```
# views.py

from django.shortcuts import render

from .models import Recipe



def recipe_list(request):

    recipes = Recipe.objects.all()

    return render(request, "recipes/list.html", {"recipes": recipes})
```

When a user visits /recipes/, Django:

1. Matches the URL to the recipe_list view in urls.py.
2. The view queries the database for all recipes.
3. Passes the data to the list.html template.
4. Renders the HTML and sends it to the browser.

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?

I would use class-based views (CBVs) because they:

- Allow code reuse through inheritance and mixins.
- Provide built-in generic views (e.g., ListView, DetailView, CreateView) to handle common patterns without writing repetitive code.
- Make it easier to extend or override specific functionality without rewriting the entire view.
  For example, in the Recipe App, we reused LoginRequiredMixin and CreateView for adding recipes with minimal code duplication.

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

Django Template Language (DTL) basics:

- **Templates are HTML files with embedded tags for dynamic content.**
- **Use {{ variable }} to display variables from the context.**
- **Use {% tag %} for logic and control structures, e.g., {% if %}, {% for %}, {% block %}.**
- **{% load static %} is used to reference static files.**
- **Filters modify variables, e.g., {{ name|upper }} converts name to uppercase.**
- **Template inheritance allows you to define a base layout and extend it in other templates using {% extends "base.html" %} and {% block content %}{% endblock %}.**
- **{% include "partial.html" %} lets you reuse template snippets.**