The Django Template System

- The Django template system is a way to dynamically generate HTML, XML, or any other text-based format using templates. Templates are HTML files with embedded Django template language (DTL) syntax.
- The template system allows you to create dynamic web pages by inserting variables, loops, conditionals, and other logic into your HTML.

Template System Basics:

- Django's template system uses double curly braces `{{ }}` to output variables and single curly braces `{% %}` for template tags, which control the logic of the template.
- Template tags can include loops (`for`), conditionals (`if`), includes (for reusing templates), and more.

Using the Template System:

- To use the template system in Django, you create templates in the `templates` directory of your app and then render these templates in your views using the `render` function.

Basic Templates and Filters:

- Templates in Django can include filters, which are used to modify variables before they are displayed. Filters are applied using the pipe (`|`) character.
- For example, `{{ variable|filter }}` would apply a filter to the variable before displaying it.

How to Configure Templates:

- Django's template system is configured in the project's settings file (`settings.py`). You can configure settings such as the template directories, context processors, and template engines.

Template Loading:

- Django uses a template loader to find and load templates from the `templates` directories of your apps. By default, Django looks for templates in the `templates` subdirectory of each app.

Template Inheritance:

- Template inheritance allows you to create a base template that contains the common elements of your site (e.g., header, footer) and then extend this base template in other templates.

- Child templates can override specific blocks of content from the parent template using the `{% block %}` tag.