

Training Day 3 Daily Dairy

june-7

Task accomplished

1. Django Templates:

- Understanding Templates:
 - Explored the concept of templates in Django for rendering dynamic HTML content.
 - Created a **templates** directory at the project level to store reusable HTML templates.
- Using Template Tags:
 - Learned to use Django template language (DTL) tags for dynamic content rendering.
 - Utilized tags like **{% for %}** and **{% if %}** for looping and conditional rendering.

Example:

html

Copy code

```
<h1>{{ title }}</h1>
<ul>
  {% for item in items %}
    <li>{{ item }}</li>
  {% endfor %}
</ul>
```



- Template Inheritance:
 - Implemented the concept of template inheritance to avoid repetition and promote reusability.
 - Created a `base.html` template with common structure (header, footer), and extended it in other templates using `{% extends %}` and `{% block %}` tags.

Example:

html

Copy code

```
<!-- base.html -->
<html>
<head>
    <title>{% block title %}My Site{%
endblock %}</title>
</head>
<body>
    <header>
        <!-- Navigation or Header -->
    </header>

    <main>
        {% block content %}{% endblock
%}
    </main>
```

```

    <footer>
        <!-- Footer -->
    </footer>
</body>
</html>

```

■

2. Views in Django:

○ Rendering Templates from Views:

- Created view functions that pass data to templates using Django's `render()` method.
- Used a context dictionary to send dynamic data from views to templates.

Example:

python

Copy code

```

from django.shortcuts import render

def homepage(request):
    context = {
        'title': 'Home Page',
        'items': ['Item 1', 'Item 2',
'Item 3']
    }

```

```
return render(request,  
'home.html', context)
```



- Handling Dynamic Data in Views:
 - Learned how to process and filter dynamic data within views before passing it to templates.
 - Practiced sending data like lists, dictionaries, and variables to be rendered dynamically on the webpage.