

Developing a Basic To-Do List Application Using Django

1. Introduction and Setup

What is Django?

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. It follows the "batteries-included" philosophy, meaning it comes with most of the essential tools built-in.

Two Features That Make Django Suitable for Web Development:

1. Built-in Admin Interface - Automatically generates an admin dashboard for managing database models.
2. ORM (Object-Relational Mapping) - Allows interaction with the database using Python code instead of SQL.

Steps to Install Django in a Virtual Environment:

1. `python -m venv env`
2. `env\Scripts\activate` (Windows) OR `source env/bin/activate` (macOS/Linux)
3. `pip install django`

2. Creating the Project

Command to create a Django project named 'todolist':

```
django-admin startproject todolist
```

Command to create a new Django app named 'tasks':

```
cd todolist
```

```
python manage.py startapp tasks
```

3. Models and Migrations

Task Model (in `tasks/models.py`):

```
from django.db import models
```

```
class Task(models.Model):
```

Developing a Basic To-Do List Application Using Django

```
title = models.CharField(max_length=200)

completed = models.BooleanField(default=False)

created_at = models.DateTimeField(auto_now_add=True)


def __str__(self):

    return self.title
```

Commands to Create and Apply Migrations:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

4. Views, Templates, and Static Files

View to Retrieve All Tasks (in tasks/views.py):

```
from django.shortcuts import render
```

```
from .models import Task
```

```
def task_list(request):

    tasks = Task.objects.all()

    return render(request, 'home.html', {'tasks': tasks})
```

Basic HTML Template (tasks/templates/home.html):

```
<!DOCTYPE html>

<html>

<head>

    <title>To-Do List</title>

    <link rel="stylesheet" href="{% static 'styles.css' %}">

</head>

<body>

    <h1>My To-Do List</h1>

    <ul>
```

Developing a Basic To-Do List Application Using Django

```
{% for task in tasks %}

<li>

    {{ task.title }} -

    {% if task.completed %}

        <span style="color: green;">Done</span>

    {% else %}

        <span style="color: red;">Pending</span>

    {% endif %}

</li>

{% endfor %}

</ul>

</body>

</html>
```

How to Link a CSS File from the Static Directory:

1. Create a 'static' folder inside the tasks app.
2. Add styles.css to it.
3. In the HTML, add {% load static %} at the top.
4. Link the CSS file using: <link rel="stylesheet" href="{% static 'styles.css' %}">