TRAINING DAY 14 REPORT

09 JULY 2025

What is a Task Manager?

A **Task Manager** is a simple web application that allows users to create, view, update, and delete their daily tasks or to-do items. It helps in organizing activities efficiently by keeping track of what needs to be done and what is already completed.

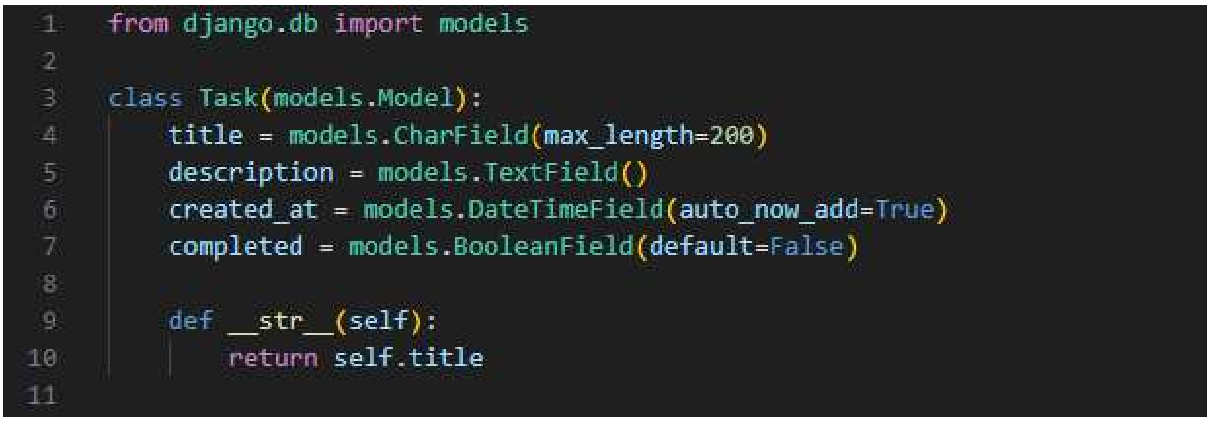
We will build a basic Django web application that:

- Stores tasks in a database.
- Displays them on a webpage.
- Allows adding new tasks, editing existing ones, and deleting tasks.

Model:

The model includes the following fields:

- **title** → CharField (to store the name or title of the task)
- **description** → TextField (to store a detailed description of the task)
- **created_at** → DateTimeField (with auto_now_add=True to automatically record the creation date and time)
- **completed** → BooleanField (default=False, to indicate whether the task is completed or still pending)



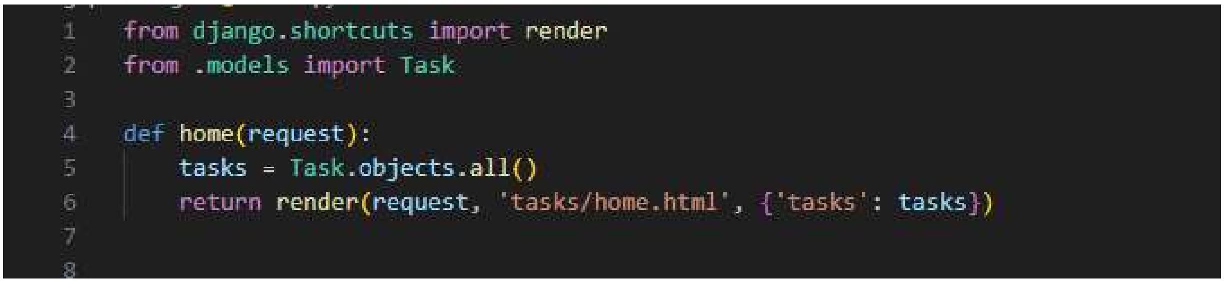
```python
from django.db import models

class Task(models.Model):
    title = models.CharField(max_length=200)
    description = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    completed = models.BooleanField(default=False)

    def __str__(self):
        return self.title

```

Steps Performed:

1. Added the Task model in the models.py file of the tasks app.
2. Created and applied migrations using the commands:
    a. python manage.py makemigrations
    b. python manage.py migrate
3. Verified that the Task table was successfully created in the SQLite database.

View Creation

- Created a view function in views.py to fetch all tasks from the database and send them to the template for display.

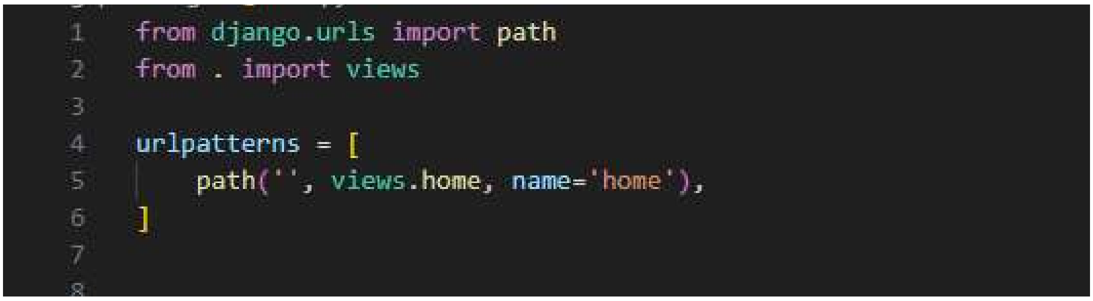


```python
from django.shortcuts import render
from .models import Task

def home(request):
    tasks = Task.objects.all()
    return render(request, 'tasks/home.html', {'tasks': tasks})
```

This view retrieves all Task objects and passes them to the home.html template inside a context dictionary.

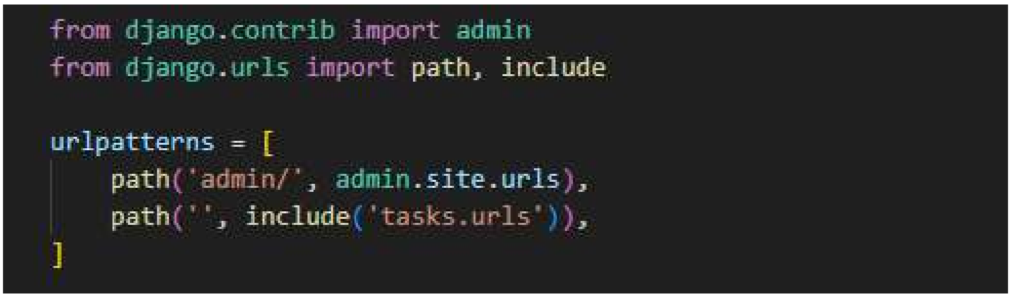
URL Configuration

- Created a urls.py file inside the tasks app and added a route for the home page.



```python
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
]
```

- Included the app’s URLs in the main project file taskmanager/urls.py:



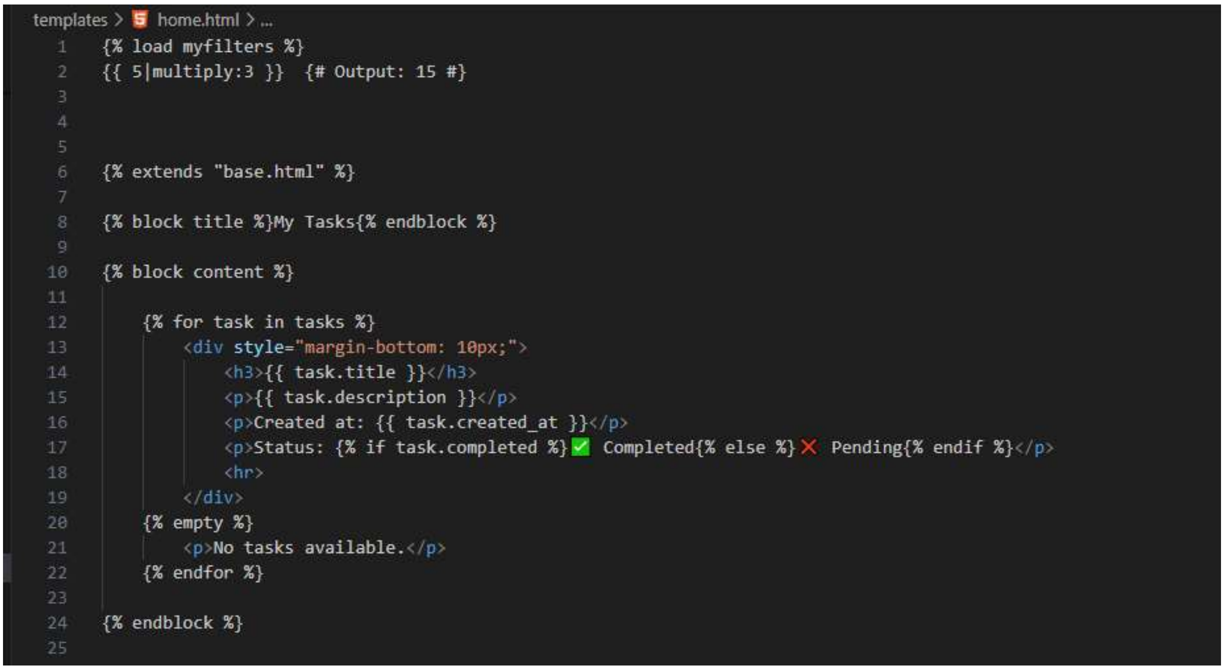
```python
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('tasks.urls')),
]
```

Template Creation

- Created a template folder structure:

  tasks/templates/tasks/home.html

- In the home.html file, displayed all tasks dynamically using Django template syntax.



```
{% load myfilters %}
{{ 5|multiply:3 }}  {# Output: 15 #}



{% extends "base.html" %}

{% block title %}My Tasks{% endblock %}

{% block content %}

    {% for task in tasks %}
        <div style="margin-bottom: 10px;">
            <h3>{{ task.title }}</h3>
            <p>{{ task.description }}</p>
            <p>Created at: {{ task.created_at }}</p>
            <p>Status: {% if task.completed %}✅ Completed{% else %}❌ Pending{% endif %}</p>
            <hr>
        </div>
    {% empty %}
        <p>No tasks available.</p>
    {% endfor %}

{% endblock %}

```

Successfully displayed all tasks from the database on the webpage, showing their title, description, creation time, and completion status.