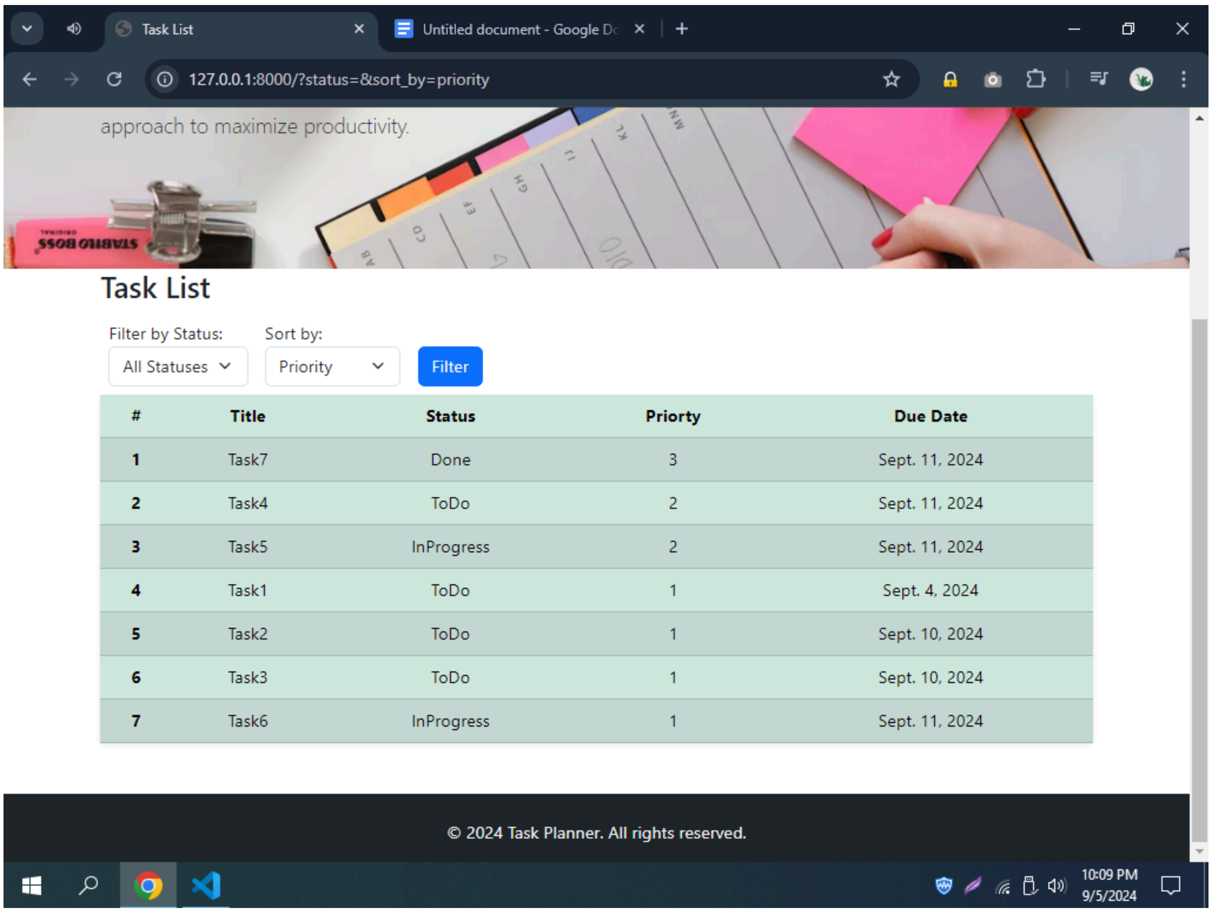
Documentation of TaskProject Assignment

- Main Screen


Task List
Untitled document - Google Dc
127.0.0.1:8000/?status=&sort_by=priority
approach to maximize productivity.
Filter by Status:
All Statuses
Sort by:
Priority
Filter
10:09 PM
9/5/2024

Task List

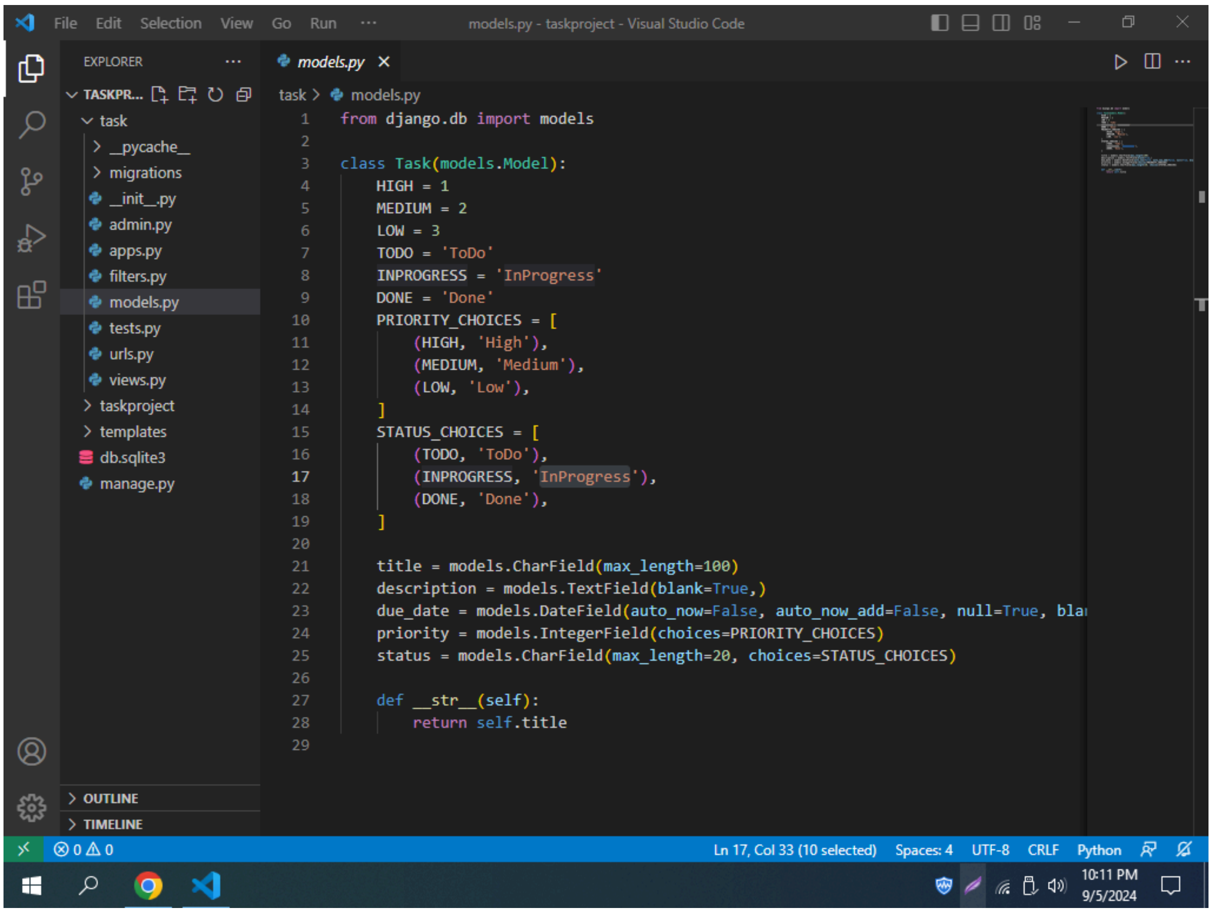
#	Title	Status	Priorty	Due Date
1	Task7	Done	3	Sept. 11, 2024
2	Task4	ToDo	2	Sept. 11, 2024
3	Task5	InProgress	2	Sept. 11, 2024
4	Task1	ToDo	1	Sept. 4, 2024
5	Task2	ToDo	1	Sept. 10, 2024
6	Task3	ToDo	1	Sept. 10, 2024
7	Task6	InProgress	1	Sept. 11, 2024

© 2024 Task Planner. All rights reserved.

We gone start with our models.py files from task app because that are defined our database schema.

Task APP

- models.py



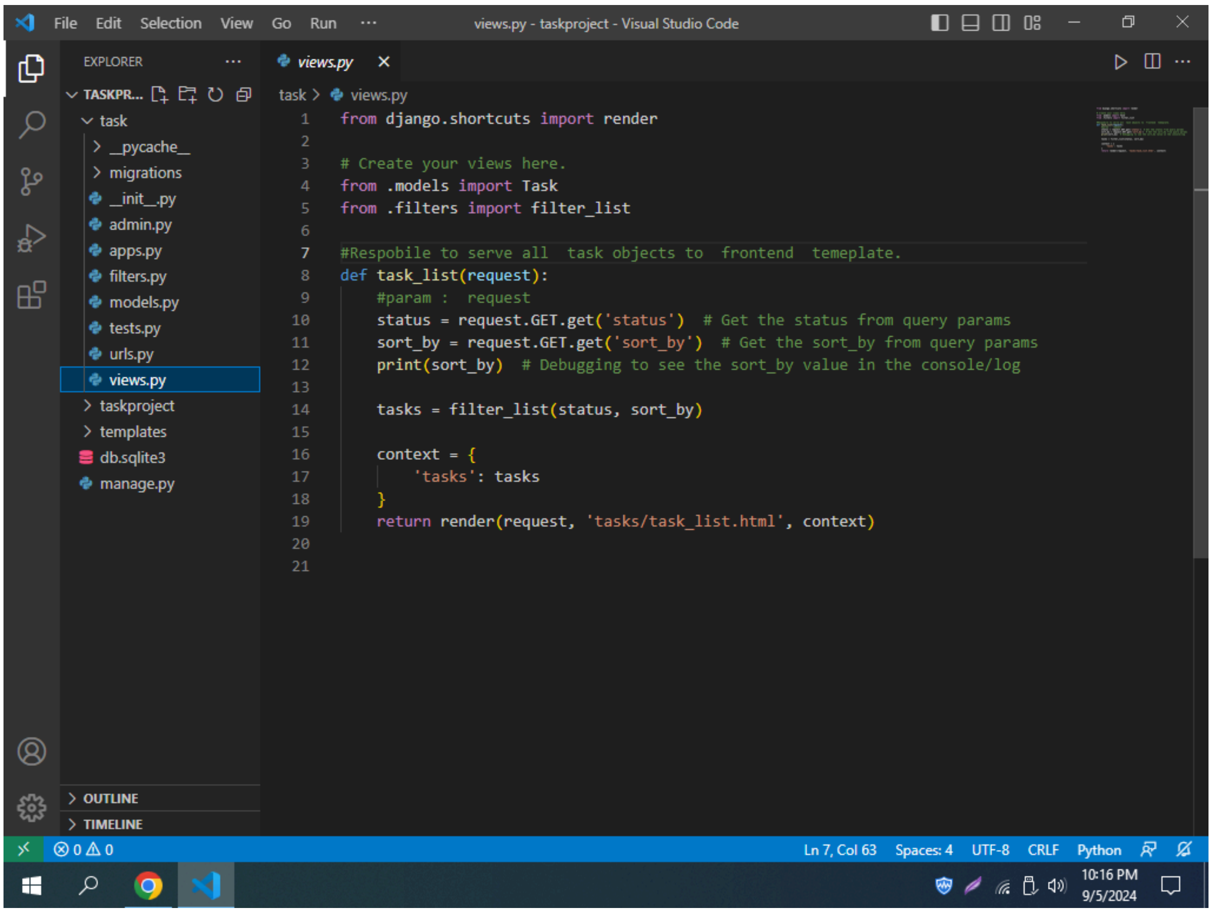
```python
from django.db import models

class Task(models.Model):
    HIGH = 1
    MEDIUM = 2
    LOW = 3
    TODO = 'ToDo'
    INPROGRESS = 'InProgress'
    DONE = 'Done'
    PRIORITY_CHOICES = [
        (HIGH, 'High'),
        (MEDIUM, 'Medium'),
        (LOW, 'Low'),
    ]
    STATUS_CHOICES = [
        (TODO, 'ToDo'),
        (INPROGRESS, 'InProgress'),
        (DONE, 'Done'),
    ]

    title = models.CharField(max_length=100)
    description = models.TextField(blank=True,)
    due_date = models.DateField(auto_now=False, auto_now_add=False, null=True, bla
    priority = models.IntegerField(choices=PRIORITY_CHOICES)
    status = models.CharField(max_length=20, choices=STATUS_CHOICES)

    def __str__(self):
        return self.title

```

Inside this we have an Task named class which contain fields that mentioned in assignment instruction like *title, description, status, due_date, priority*
And each data type is like this

- Title: str
- Description: str
- Due_date : date
- Priority:str
- Status:str

priority and status is string type because that are choices kind fields that have to store value according user choices

Now we gone look's our views that responsible for our Main Screen serving and and our filter to works so for that we have to go inside *views.py* file of Task app views.py



```python
from django.shortcuts import render

# Create your views here.
from .models import Task
from .filters import filter_list

#Respobile to serve all  task objects to  frontend  temeplate.
def task_list(request):
    #param :  request
    status = request.GET.get('status')  # Get the status from query params
    sort_by = request.GET.get('sort_by')  # Get the sort_by from query params
    print(sort_by)  # Debugging to see the sort_by value in the console/log

    tasks = filter_list(status, sort_by)

    context = {
        'tasks': tasks
    }
    return render(request, 'tasks/task_list.html', context)
```

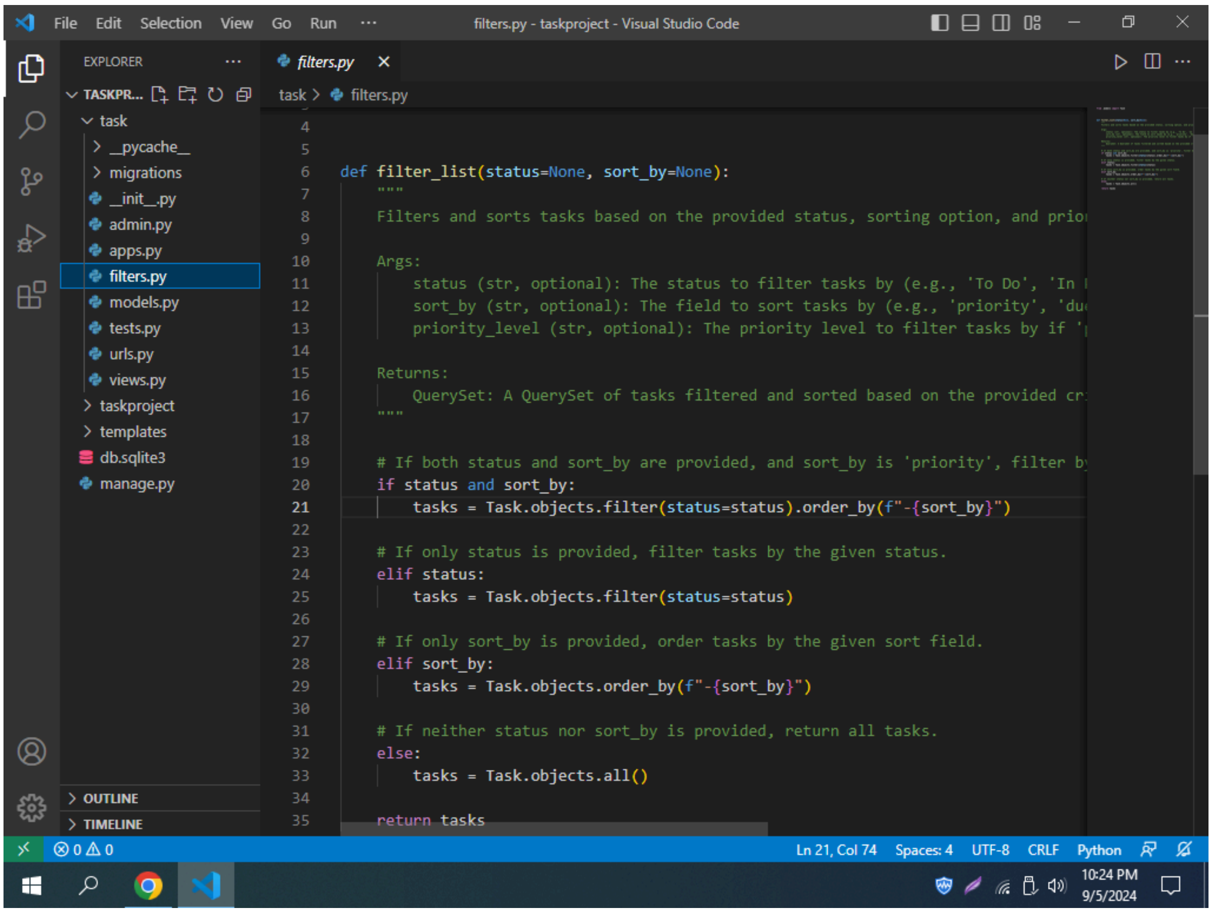
Let's break down this code. Inside this we have a function named task_list this function has to serve our main screen to the user and handle all operations related to our ***filter and sorting*** and return with correct data as task objects.

Inside this our only parameter is request which is common parameter to served an ***html or http in django***

And another two line ***status, sort_by*** things that we get from html page for filtering and sorting tasks according to these values it only triggers when the main screen user clicks on that ***filter button.***

And the main things comes tasks this line passing all that selected values from sort_by and status our main function that actually filter that objects that is ***filter_list() function***

- filters.py



```python

def filter_list(status=None, sort_by=None):
    """
    Filters and sorts tasks based on the provided status, sorting option, and prio

    Args:
        status (str, optional): The status to filter tasks by (e.g., 'To Do', 'In 
        sort_by (str, optional): The field to sort tasks by (e.g., 'priority', 'du
        priority_level (str, optional): The priority level to filter tasks by if '

    Returns:
        QuerySet: A QuerySet of tasks filtered and sorted based on the provided cr
    """

    # If both status and sort_by are provided, and sort_by is 'priority', filter b
    if status and sort_by:
        tasks = Task.objects.filter(status=status).order_by(f"-{sort_by}")

    # If only status is provided, filter tasks by the given status.
    elif status:
        tasks = Task.objects.filter(status=status)

    # If only sort_by is provided, order tasks by the given sort field.
    elif sort_by:
        tasks = Task.objects.order_by(f"-{sort_by}")

    # If neither status nor sort_by is provided, return all tasks.
    else:
        tasks = Task.objects.all()

    return tasks
```

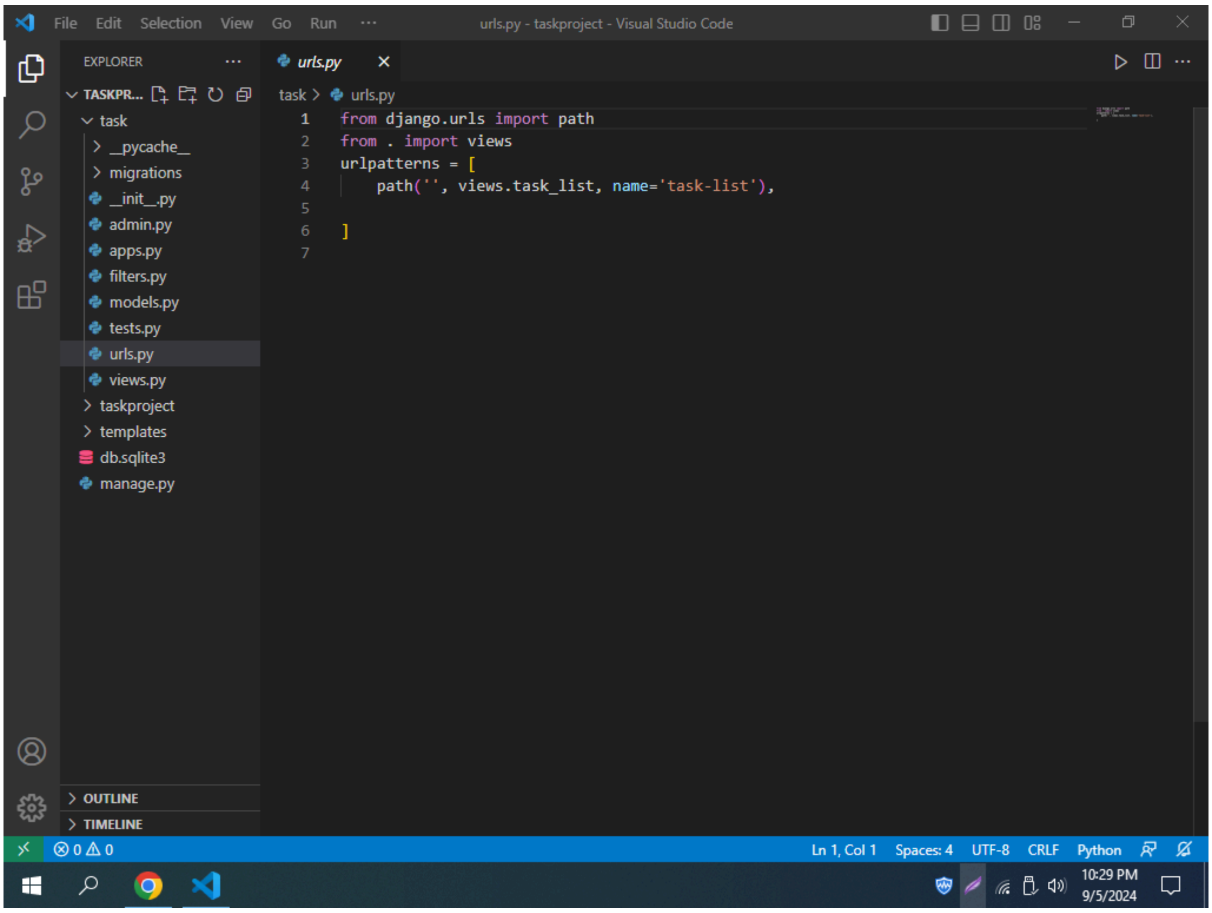
Inside this we getting only one function that handles all our filter and sorting operations

Parameter: <u>status , sort_by</u>

Initially this both parameter is None because if user have to gets all objects

Here We handle different scenarios for sorting and filter where some time can want to use both sort and filter or some time only one.

Inside this function we are using the ***django ORM method***, *filter()* and for sorting we use *order_by()* in descending order.

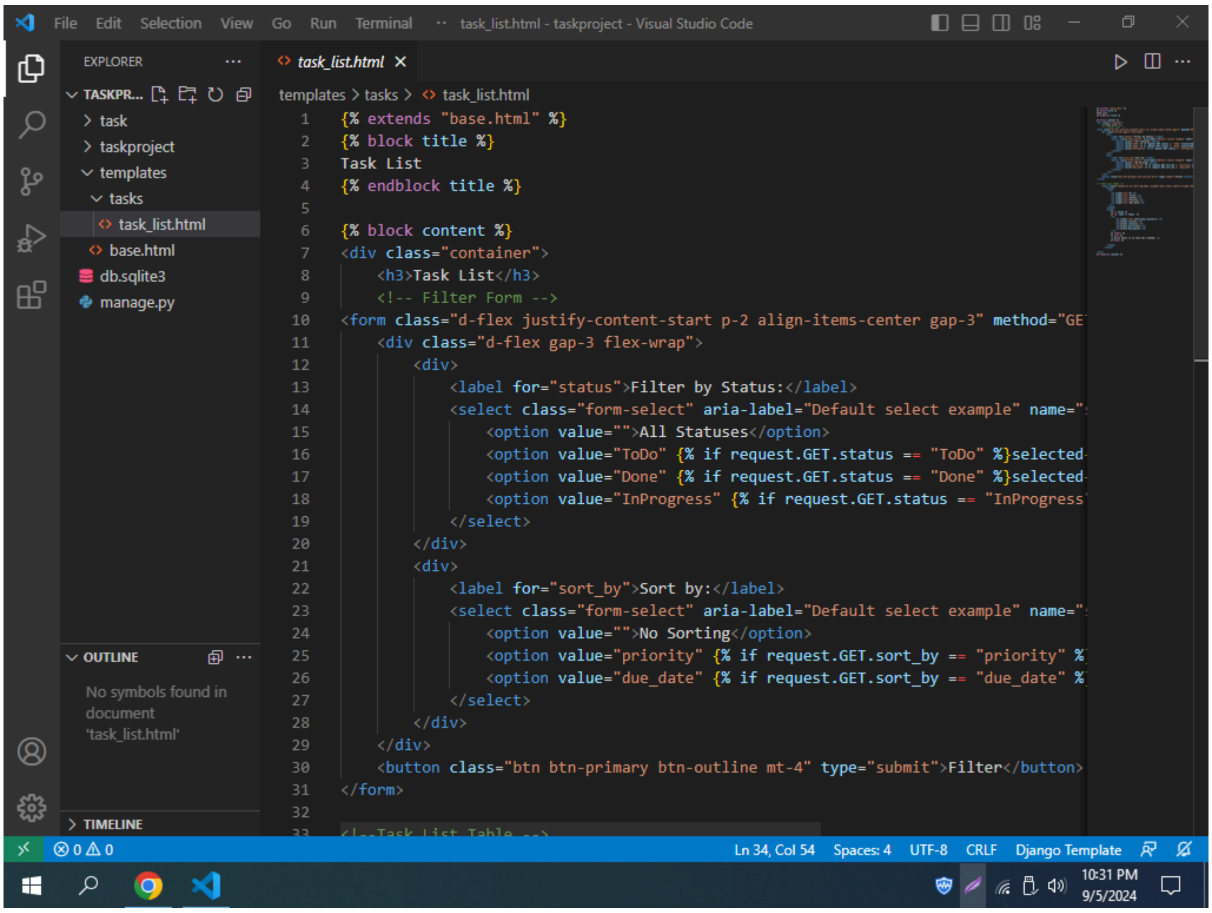


```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.task_list, name='task-list'),

]
```

- urls.py

This is simply just routing of our task web app as root because here we do not passing any point

Now we gone look inside templates code



```
{% extends "base.html" %}
{% block title %}
Task List
{% endblock title %}

{% block content %}
<div class="container">
    <h3>Task List</h3>
    <!-- Filter Form -->
<form class="d-flex justify-content-start p-2 align-items-center gap-3" method="GE
    <div class="d-flex gap-3 flex-wrap">
        <div>
            <label for="status">Filter by Status:</label>
            <select class="form-select" aria-label="Default select example" name="
                <option value="">All Statuses</option>
                <option value="ToDo" {% if request.GET.status == "ToDo" %}selected
                <option value="Done" {% if request.GET.status == "Done" %}selected
                <option value="InProgress" {% if request.GET.status == "InProgress
            </select>
        </div>
        <div>
            <label for="sort_by">Sort by:</label>
            <select class="form-select" aria-label="Default select example" name="
                <option value="">No Sorting</option>
                <option value="priority" {% if request.GET.sort_by == "priority" %
                <option value="due_date" {% if request.GET.sort_by == "due_date" %
            </select>
        </div>
    </div>
    <button class="btn btn-primary btn-outline mt-4" type="submit">Filter</button>
</form>

<!--Task List Table -->
```

- task_list.html

Inside this html file we just just and form and table table just displaying returned task objects fromtask_list function of views and form is responsible for filtering things