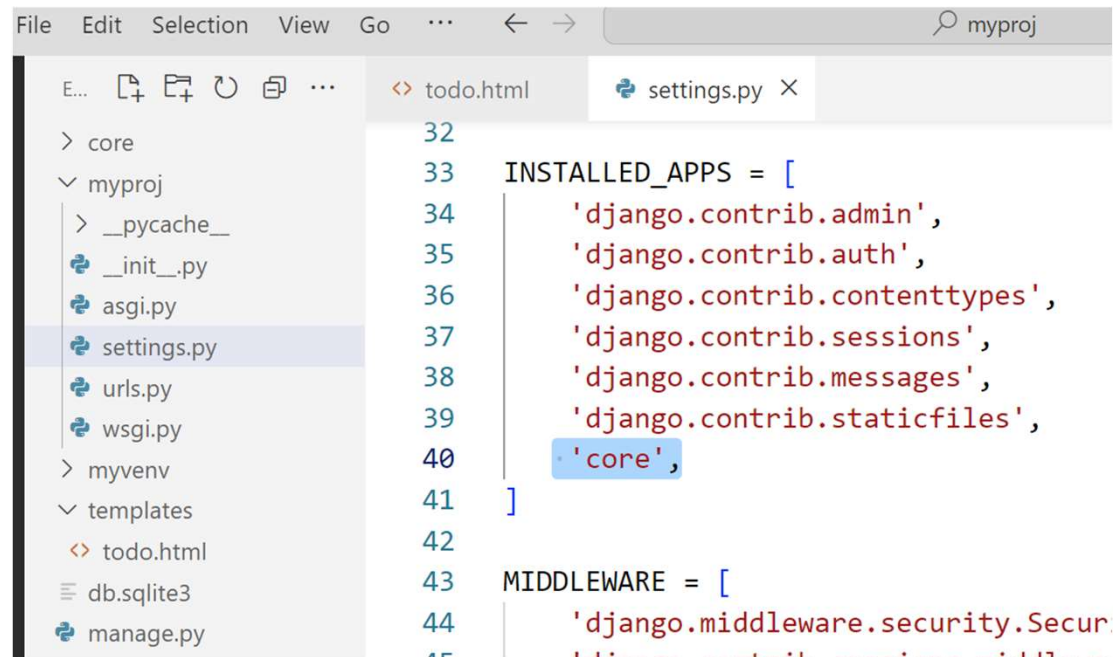


Configure core in settings.py

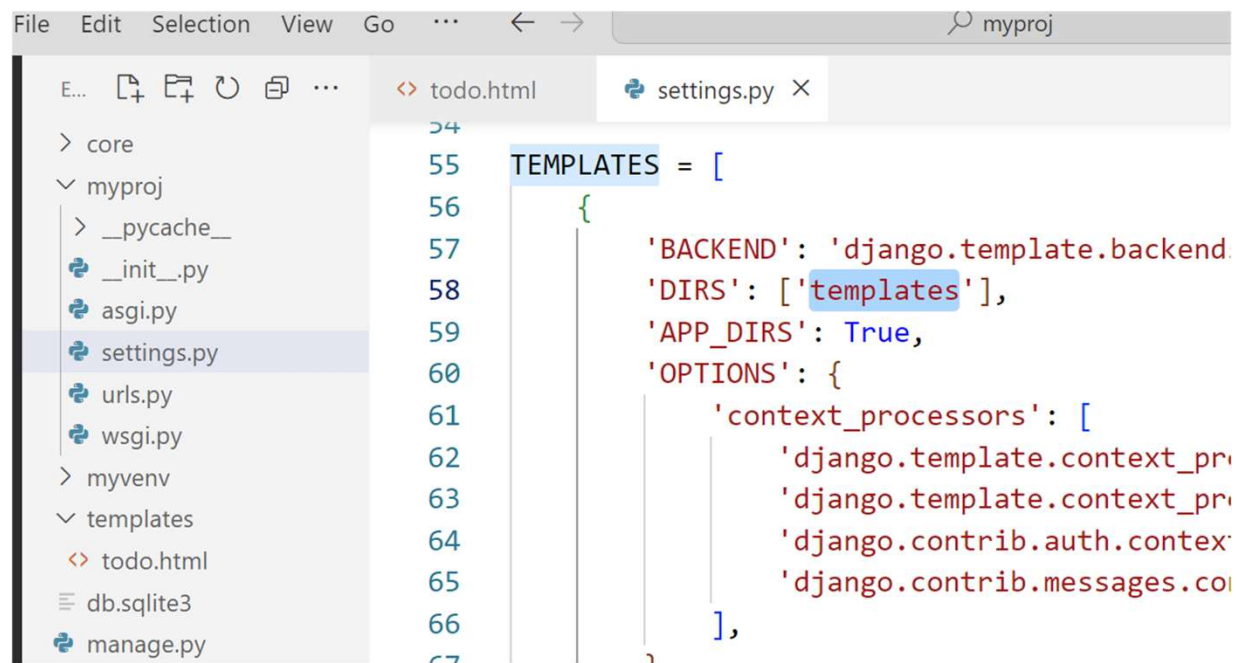


```
File Edit Selection View Go ... < > myproj

> core
✓ myproj
  > __pycache__
  __init__.py
  asgi.py
  settings.py
  urls.py
  wsgi.py
> myenv
✓ templates
  <> todo.html
  db.sqlite3
  manage.py

32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40     'core',
41 ]
42
43 MIDDLEWARE = [
44     'django.middleware.security.SecurityMiddleware',
45     'django.contrib.sessions.middleware.SessionMiddleware',
46     'django.middleware.common.CommonMiddleware',
47     'django.middleware.csrf.CsrfViewMiddleware',
48     'django.contrib.auth.middleware.AuthenticationMiddleware',
49     'django.contrib.messages.middleware.MessageMiddleware',
50 ]
```

Configure templates in settings.py



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a 'myproj' directory containing 'settings.py', 'urls.py', 'wsgi.py', and 'manage.py'. The 'settings.py' file is selected. The code editor shows the 'TEMPLATES' configuration in 'settings.py'.

```
File Edit Selection View Go ... < > myproj
E... [Icons] ... <> todo.html settings.py X
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': ['templates'],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.login_required',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     ],
69 ]
```

Create todo model

Update models.py

```
from django.db import models  
class todo(models.Model):  
    task=models.CharField(max_length=50)
```

Run following in terminal:

```
python manage.py makemigrations  
python manage.py migrate
```

Create todo.html inside templates

```
<!DOCTYPE html>
<html lang="en">
  <head><title>Document</title></head>
  <body>
    <h1>Todo Application</h1>
    <form method="POST">
      {% csrf_token %} Cross-Site Request Forgery
      <p><input type="text" name="task" /><button>Add</button></p>
    </form>
    {% for row in todos%}
      <div>
        {{row.task}} <a href="/deleteTodo/?id={{row.id}}">Delete</a>
      </div>
    {% endfor %}
  </body>
</html>
```

Add todo function in views.py

```
from core import models

def todo(request):
    if request.method=="POST":
        task = request.POST.get("task")
        c = models.todo(task=task)
        c.save()
    mytodo = models.todo.objects.all()
    data = {'todos':mytodo}
    return render(request,"todo.html",data)
```

Add deleteTodo function in views.py

```
def deleteTodo(request):  
    if request.method=="GET":  
        todold = int(request.GET.get("id"))  
        models.todo.objects.filter(id=todold).delete()  
    return todo(request)
```

Add path in urls.py

```
from django.contrib import admin
from django.urls import path,include
from core import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',views.todo),
    path('deleteTodo/',views.deleteTodo),
]
```

Run server and view todo in browser

In terminal type > `python manage.py runserver`

open any browser and type following url to view todo app

<http://127.0.0.1:8000/>