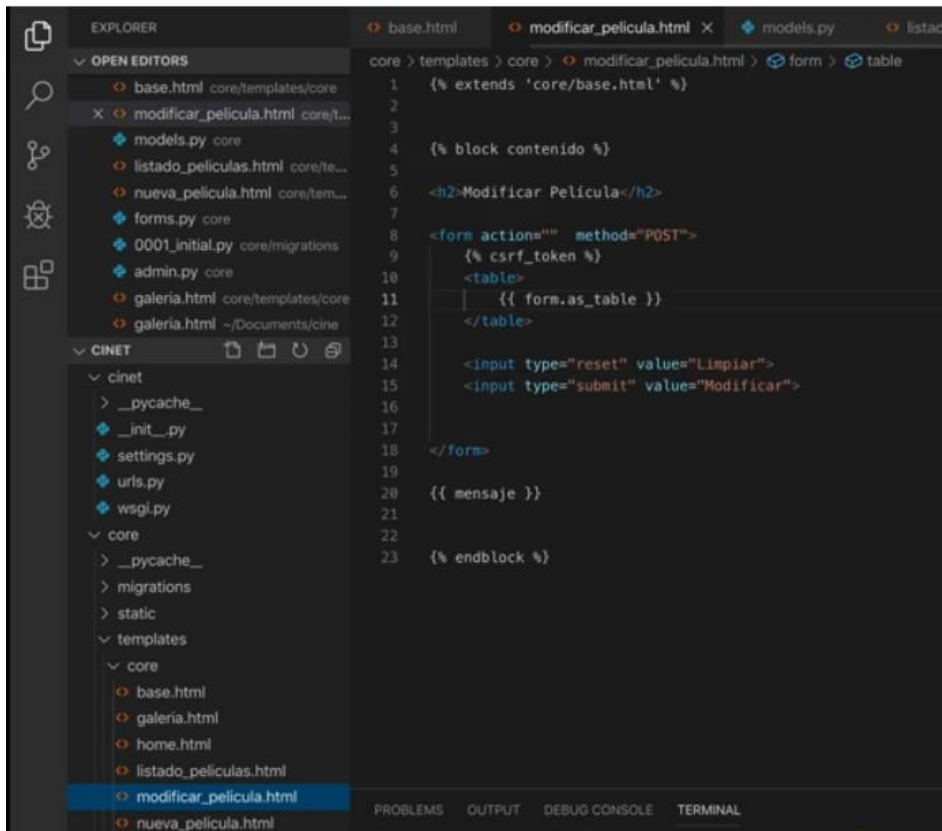


Para modificar y eliminar:

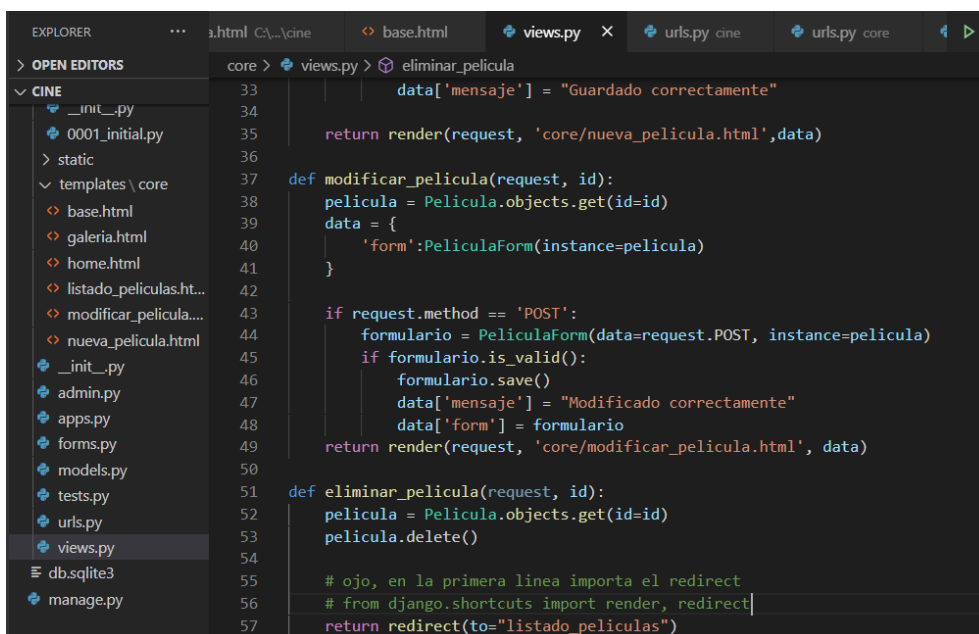
Primero debes crear: `core/templates/core/modificar_pelicula.html`



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure, including the `core/templates/core` directory. The `modificar_pelicula.html` file is highlighted in the list. The main editor window shows the content of this file, which extends the `core/base.html` template. The code includes a block for the page content, an `h2` header for 'Modificar Pelicula', a form with a CSRF token, a table for the form, and two input fields for 'Limpiar' and 'Modificar'.

```
1 {% extends 'core/base.html' %}
2
3
4 {% block contenido %}
5
6 <h2>Modificar Pelicula</h2>
7
8 <form action="" method="POST">
9     {% csrf_token %}
10     <table>
11         {{ form.as_table }}
12     </table>
13
14     <input type="reset" value="Limpiar">
15     <input type="submit" value="Modificar">
16
17 </form>
18
19 {{ mensaje }}
20
21
22 {% endblock %}
```

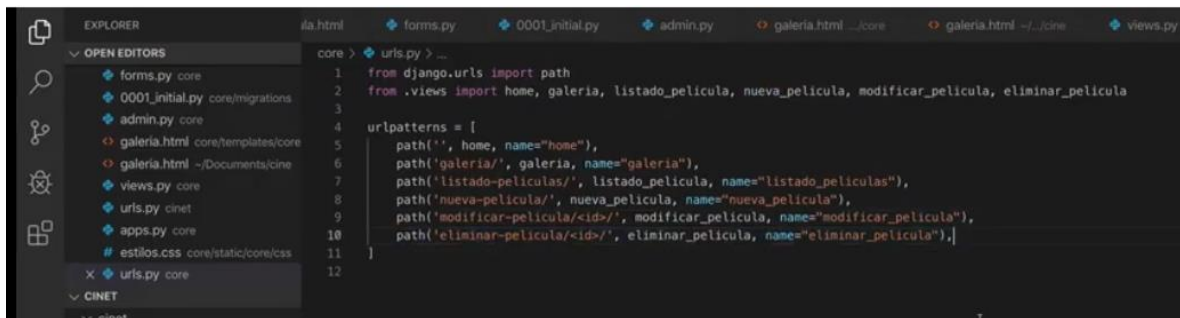
Después agregas los métodos en `views.py`



The screenshot shows the Visual Studio Code interface with the `views.py` file open. The code defines two methods: `modificar_pelicula` and `eliminar_pelicula`. The `modificar_pelicula` method handles the POST request, validates the form, and saves the changes. The `eliminar_pelicula` method handles the DELETE request and deletes the movie record. Both methods use Django's `render` and `redirect` functions.

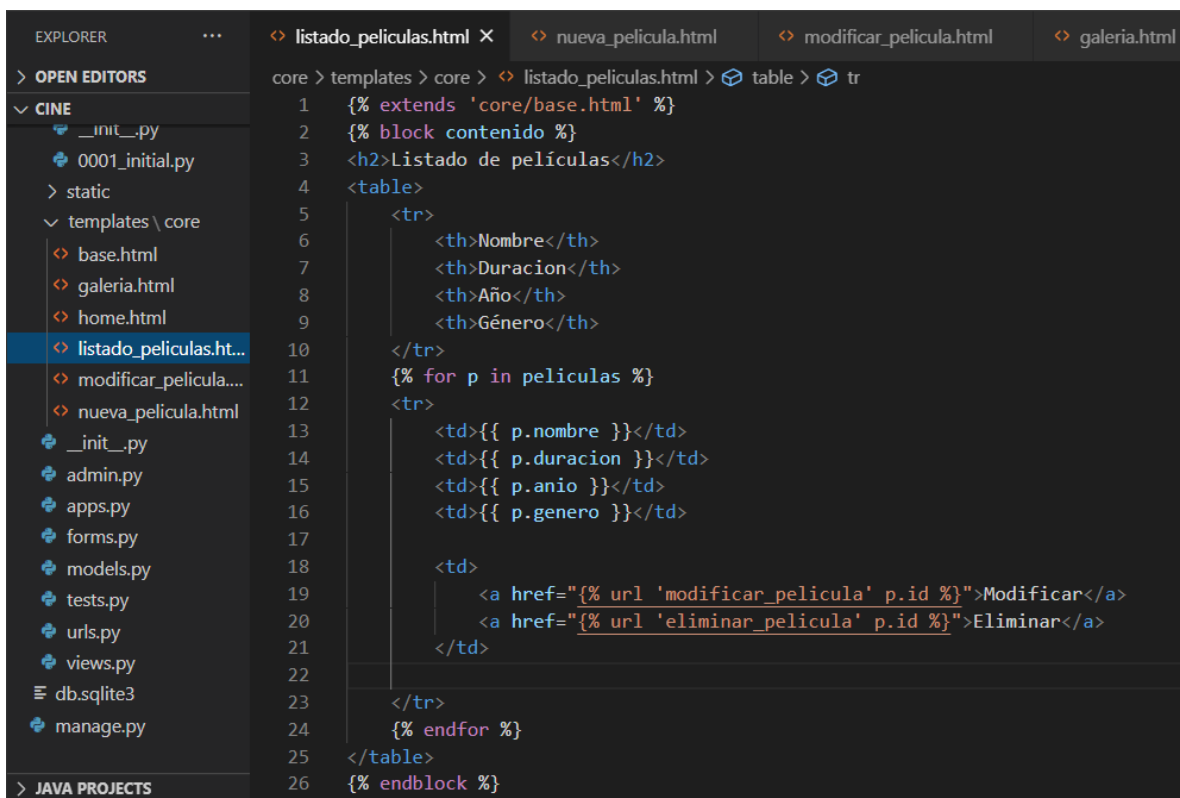
```
33 data['mensaje'] = "Guardado correctamente"
34
35 return render(request, 'core/nueva_pelicula.html', data)
36
37 def modificar_pelicula(request, id):
38     pelicula = Pelicula.objects.get(id=id)
39     data = {
40         'form': PeliculaForm(instance=pelicula)
41     }
42
43     if request.method == 'POST':
44         formulario = PeliculaForm(data=request.POST, instance=pelicula)
45         if formulario.is_valid():
46             formulario.save()
47             data['mensaje'] = "Modificado correctamente"
48             data['form'] = formulario
49     return render(request, 'core/modificar_pelicula.html', data)
50
51 def eliminar_pelicula(request, id):
52     pelicula = Pelicula.objects.get(id=id)
53     pelicula.delete()
54
55     # ojo, en la primera linea importa el redirect
56     # from django.shortcuts import render, redirect
57     return redirect(to="listado_peliculas")
```

Paso 3, Debes dar el permiso en urls.py para que funcione el navegador



```
1 from django.urls import path
2 from .views import home, galeria, listado_pelicula, nueva_pelicula, modificar_pelicula, eliminar_pelicula
3
4 urlpatterns = [
5     path('', home, name="home"),
6     path('galeria/', galeria, name="galeria"),
7     path('listado-peliculas/', listado_pelicula, name="listado_peliculas"),
8     path('nueva-pelicula/', nueva_pelicula, name="nueva_pelicula"),
9     path('modificar-pelicula/<id>/', modificar_pelicula, name="modificar_pelicula"),
10    path('eliminar-pelicula/<id>/', eliminar_pelicula, name="eliminar_pelicula"),
11 ]
12
```

RECUERDA DE MODIFICAR EL listado_peliculas.html para que te muestre los link que permiten modificar y/o eliminar un registro



```
1 {% extends 'core/base.html' %}
2 {% block contenido %}
3 <h2>Listado de películas</h2>
4 <table>
5     <tr>
6         <th>Nombre</th>
7         <th>Duracion</th>
8         <th>Año</th>
9         <th>Género</th>
10    </tr>
11    {% for p in peliculas %}
12    <tr>
13        <td>{{ p.nombre }}</td>
14        <td>{{ p.duracion }}</td>
15        <td>{{ p.anio }}</td>
16        <td>{{ p.genero }}</td>
17
18        <td>
19            <a href="{% url 'modificar_pelicula' p.id %}">Modificar</a>
20            <a href="{% url 'eliminar_pelicula' p.id %}">Eliminar</a>
21        </td>
22    </tr>
23    {% endfor %}
24 </table>
25 {% endblock %}
```



Listado de películas

Nombre	Duracion	Año	Género	
Los intocables	135	1985	Acción	Modificar Eliminar
El Cachalote	199	2000	Terror	Modificar Eliminar
Pancho Pistolas	120	1960	Comedia	Modificar Eliminar