

# Django Watchlist Projekt mit Accounts und CSS

---

Dieses Projekt umfasst:

1. Benutzerregistrierung, Login und Logout in einer separaten **accounts**-App.
2. Eine **movies**-App für die Verwaltung einer persönlichen Watchlist.
3. Eine CSS-Datei für das Layout und Styling.
4. Zugriffsbeschränkungen, damit Benutzer nur auf ihre eigenen Inhalte zugreifen können.
5. Zwei Änderungen:
  - **Benutzer werden zur Login-Seite weitergeleitet**, wenn sie versuchen, geschützte Seiten aufzurufen, ohne angemeldet zu sein.
  - Das **Movie**-Modell wurde erweitert, um eine **IMDb-URL** zu speichern.

---

## 1. Projektstruktur

```
watchlist_project/
├── accounts/ (für Benutzerverwaltung)
│   ├── forms.py (Registrierungsformular)
│   ├── urls.py (Routen für Benutzerverwaltung)
│   └── views.py (Logik für Registrierung, Login und Logout)
├── movies/ (für Watchlist-Verwaltung)
│   ├── forms.py (Film-Formular)
│   ├── models.py (Film-Modell mit Benutzerassoziation)
│   ├── urls.py (Routen für Watchlist-Management)
│   └── views.py (Logik für Watchlist und Film-Erstellung)
├── templates/ (Base-Template und spezifische Seiten)
├── static/css/ (CSS-Datei für Styling)
├── manage.py (Projektverwaltung)
└── watchlist_project/ (Hauptprojekt mit Einstellungen und URLs)
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Django Watchlist{% endblock %}</title>
  <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="{% url 'movies:watchlist' %}">Meine Watchlist</a></li>
        <li><a href="{% url 'movies:add_movie' %}">Film hinzufügen</a></li>
        {% if user.is_authenticated %}
          <li><a href="{% url 'accounts:logout' %}">Abmelden</a></li>
        {% else %}
          <li><a href="{% url 'accounts:login' %}">Anmelden</a></li>
          <li><a href="{% url 'accounts:register' %}">Registrieren</a></li>
        {% endif %}
        {% if user.is_staff %}
          <li><a href="/admin/">Admin-Bereich</a></li>
        {% endif %}
      </ul>
    </nav>
  </header>

  <main>
```

```
        {% block content %}{% endblock %}
    </main>

    <footer>
        <p>&copy; 2024 Django Watchlist. Alle Rechte vorbehalten.</p>
    </footer>
</body>
</html>
```

---

## 2. Accounts-App für Benutzerverwaltung

### URLs in der **accounts**-App

#### **accounts/urls.py**

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

app_name = 'accounts'

urlpatterns = [
    path('register/', views.register, name='register'),
    path('login/', views.login_view, name='login'),
    path('logout/', views.logout_view, name='logout'),
]
```

### Views in der **accounts**-App

#### **accounts/views.py**

```
from django.shortcuts import render, redirect
from django.contrib.auth import login, logout
from django.contrib.auth.forms import AuthenticationForm
from .forms import UserRegistrationForm
from django.contrib import messages

def register(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            user = form.save(commit=False)
            user.set_password(form.cleaned_data['password'])
            user.save()
            messages.success(request, 'Registrierung erfolgreich! Bitte melden Sie sich an.')
            return redirect('accounts:login')
    else:
        form = UserRegistrationForm()
        return render(request, 'accounts/register.html', {'form': form})

def login_view(request):
    if request.method == 'POST':
        form = AuthenticationForm(data=request.POST)
        if form.is_valid():
            user = form.get_user()
            login(request, user)
            return redirect('movies:watchlist')
    else:
        form = AuthenticationForm()
        return render(request, 'accounts/login.html', {'form': form})

def logout_view(request):
```

```
logout(request)
return redirect('accounts:login')
```

## Forms für Benutzerregistrierung

accounts/forms.py

```
from django import forms
from django.contrib.auth.models import User

class UserRegistrationForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput)
    confirm_password = forms.CharField(widget=forms.PasswordInput)

    class Meta:
        model = User
        fields = ['username', 'email', 'password']

    def clean(self):
        cleaned_data = super().clean()
        password = cleaned_data.get('password')
        confirm_password = cleaned_data.get('confirm_password')

        if password != confirm_password:
            raise forms.ValidationError("Passwörter stimmen nicht überein.")
```

## Templates für Accounts

templates/accounts/register.html

```
{% extends "base.html" %}

{% block content %}
<h2>Registrierung</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Registrieren</button>
</form>
{% endblock %}
```

templates/accounts/login.html

```
{% extends "base.html" %}

{% block content %}
<h2>Anmeldung</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Anmelden</button>
</form>
{% endblock %}
```

---

## 3. Movies-App für die Watchlist

### Movie-Modell

## movies/models.py

```
from django.db import models
from django.contrib.auth.models import User

class Movie(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    title = models.CharField(max_length=255)
    director = models.CharField(max_length=255)
    release_year = models.PositiveIntegerField()
    genre = models.CharField(max_length=100)
    imdb_url = models.URLField(blank=True, null=True) # IMDb-URL optional

    def __str__(self):
        return self.title
```

## Views für Movies

### movies/views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from .models import Movie
from .forms import MovieForm

@login_required(login_url='/accounts/login/')
def watchlist_view(request):
    movies = Movie.objects.filter(user=request.user)
    return render(request, 'movies/watchlist.html', {'movies': movies})

@login_required(login_url='/accounts/login/')
def add_movie(request):
    if request.method == 'POST':
        form = MovieForm(request.POST)
        if form.is_valid():
            movie = form.save(commit=False)
            movie.user = request.user
            movie.save()
            return redirect('movies:watchlist')
    else:
        form = MovieForm()
    return render(request, 'movies/add_movie.html', {'form': form})
```

## Formular für Movies

### movies/forms.py

```
from django import forms
from .models import Movie

class MovieForm(forms.ModelForm):
    class Meta:
        model = Movie
        fields = ['title', 'director', 'release_year', 'genre', 'imdb_url']
```

## Templates für Movies

### templates/movies/watchlist.html

```

{% extends "base.html" %}

{% block content %}
<h2>Deine Watchlist</h2>
<ul>
  {% for movie in movies %}
    <li>
      <strong>{{ movie.title }}</strong> ({{ movie.release_year }}) - {{ movie.genre }}
      <br>Regisseur: {{ movie.director }}
      {% if movie.imdb_url %}
        <br><a href="{{ movie.imdb_url }}" target="_blank">IMDb-Seite</a>
      {% endif %}
    </li>
  {% endfor %}
</ul>
<a href="{% url 'movies:add_movie' %}">Film hinzufügen</a>
{% endblock %}

```

### templates/movies/add\_movie.html

```

{% extends "base.html" %}

{% block content %}
<h2>Film hinzufügen</h2>
<form method="post">
  {% csrf_token %}
  {{ form.as_p }}
  <button type="submit">Hinzufügen</button>
</form>
{% endblock %}

```

## 4. CSS-Datei

### static/css/styles.css

```

body {
  font-family: Arial, sans-serif;
  background-color: #f9f9f9;
  margin: 0;
  padding: 0;
}

header {
  background-color: #333;
  color: #fff;
  padding: 15px;
  text-align: center;
}

header a {
  color: #fff;
  text-decoration: none;
  margin: 0 10px;
}

header a:hover {
  text-decoration: underline;
}

main {

```

```
margin: 20px;
padding: 20px;
background: #fff;
border: 1px solid #ddd;
border-radius: 8px;
}

footer {
background-color: #333;
color: #fff;
text-align: center;
padding: 10px 0;
position: fixed;
bottom: 0;
width: 100%;
}

button {
background-color: #007bff;
color: #fff;
border: none;
padding: 10px 15px;
cursor: pointer;
border-radius: 5px;
}

button:hover {
background-color: #0056b3;
}
```

---

## 5. URLs im Projekt

`watchlist_project/urls.py`

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

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

---

## Abschluss

Mit diesen Änderungen können Benutzer:

- sich registrieren und anmelden,
- nur ihre eigenen Inhalte sehen,
- Filme mit einer IMDb-URL hinzufügen,
- und geschützte Seiten erfordern eine Anmeldung.