**Лабораторная работа 4: Разработка пользовательского интерфейса и взаимодействие с API**

**Цель работы:**  
Создание клиентского веб-приложения с использованием Django, подключение его к серверу через REST API и реализация основных пользовательских сценариев.

**1. Описание структуры интерфейса**

Приложение реализует интерфейс онлайн-кинотеатра. Пользователь может:

* Зарегистрироваться и войти в аккаунт (страницы регистрации и авторизации).
* Просматривать список фильмов и сериалов.
* Переходить на отдельные страницы фильмов, сезонов, эпизодов.
* Оставлять отзывы к контенту (если авторизован).
* Видеть отзывы других пользователей.

**Основные страницы:**

* /register/ — страница регистрации.
* /login/ — страница входа.
* /films/ и /series/ — списки фильмов и сериалов.
* /films/<id>/ и /series/<id>/ — детальная страница фильма или сериала.
* /series/<id>/seasons/<id>/episodes/<id>/ — страница эпизода с отзывами.

Все страницы реализованы с использованием Django-шаблонов (Jinja2) и обрабатываются сервером.

**2. Код макетов (шаблонов)**

Примеры шаблонов:

**templates/login.html**

<form method="post" action="/login/">

{% csrf\_token %}

<input type="text" name="username" placeholder="Username">

<input type="password" name="password" placeholder="Password">

<button type="submit">Login</button>

</form>

**templates/register.html**

<form method="post" action="/register/">

{% csrf\_token %}

<input type="text" name="username" placeholder="Username">

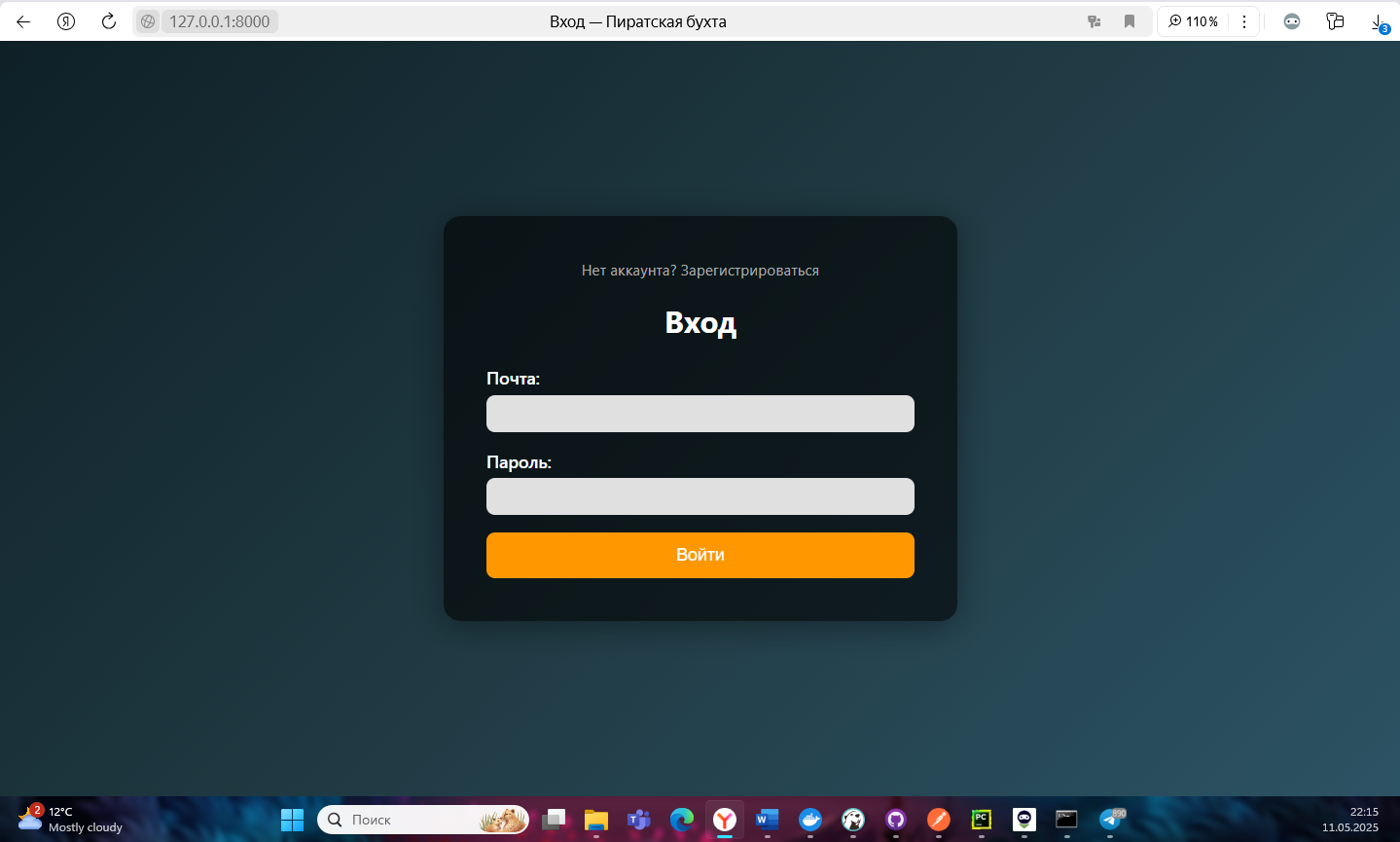
<input type="password" name="password" placeholder="Password">

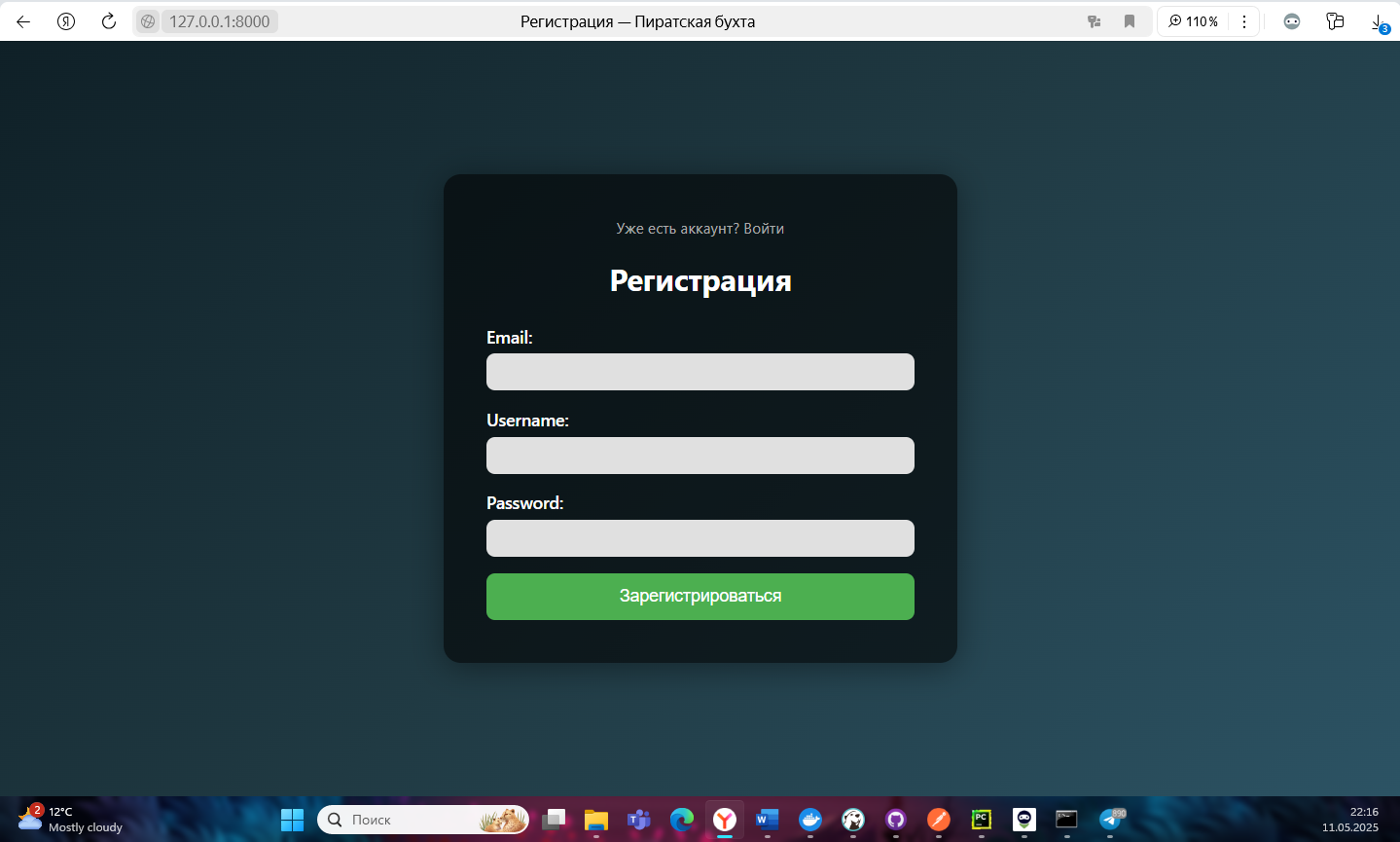
<input type="email" name="email" placeholder="Email">

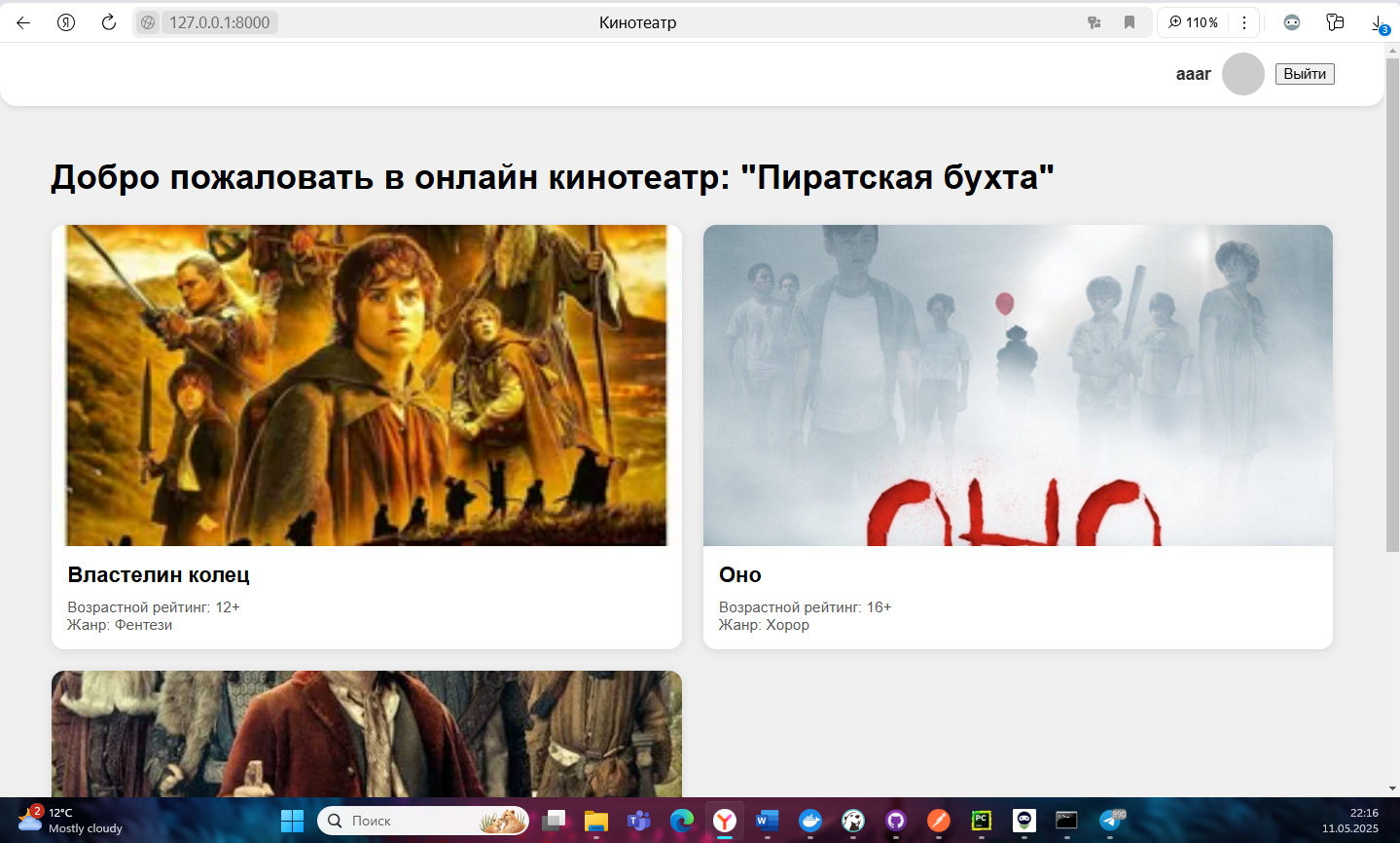
<button type="submit">Register</button>

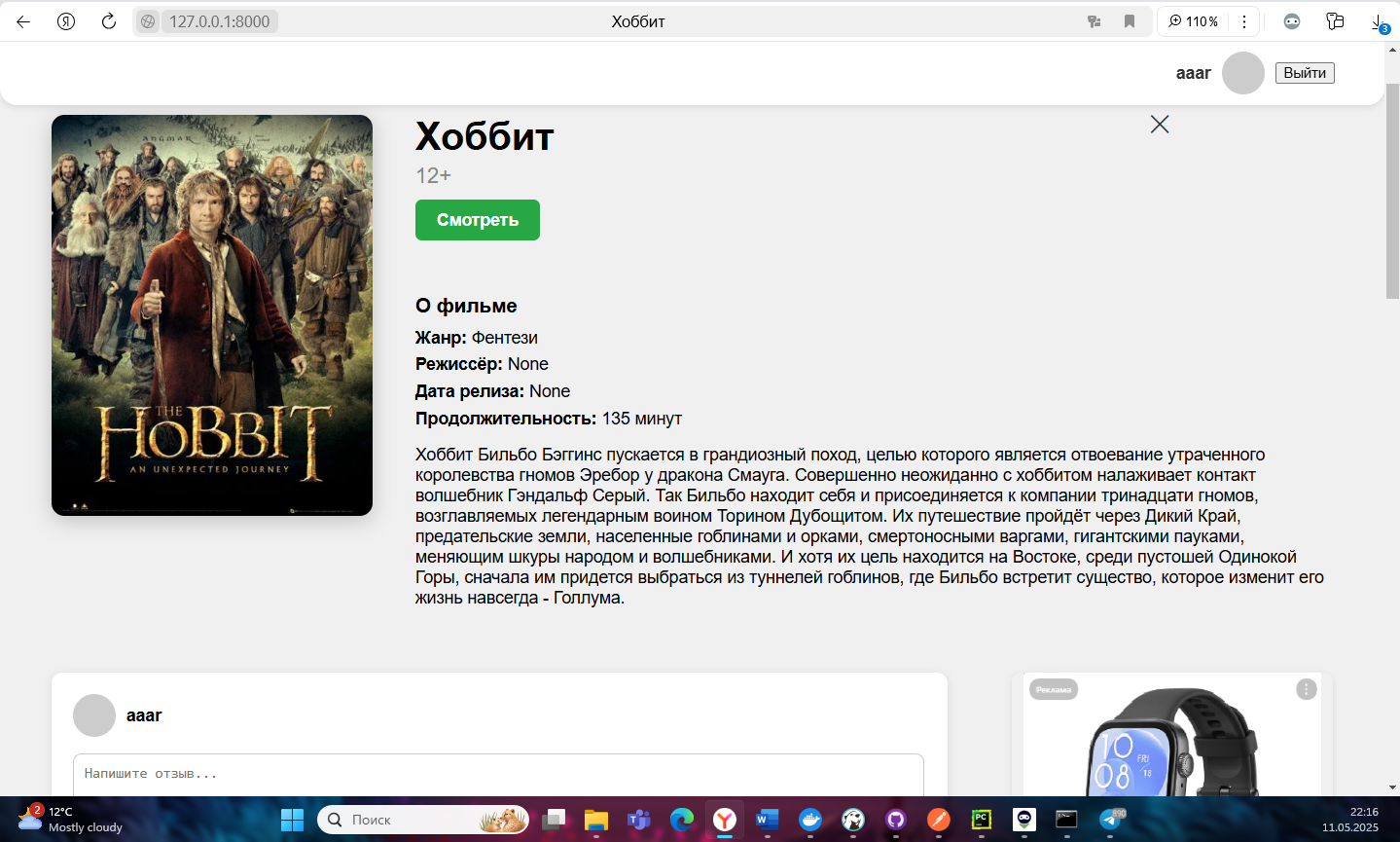
</form>

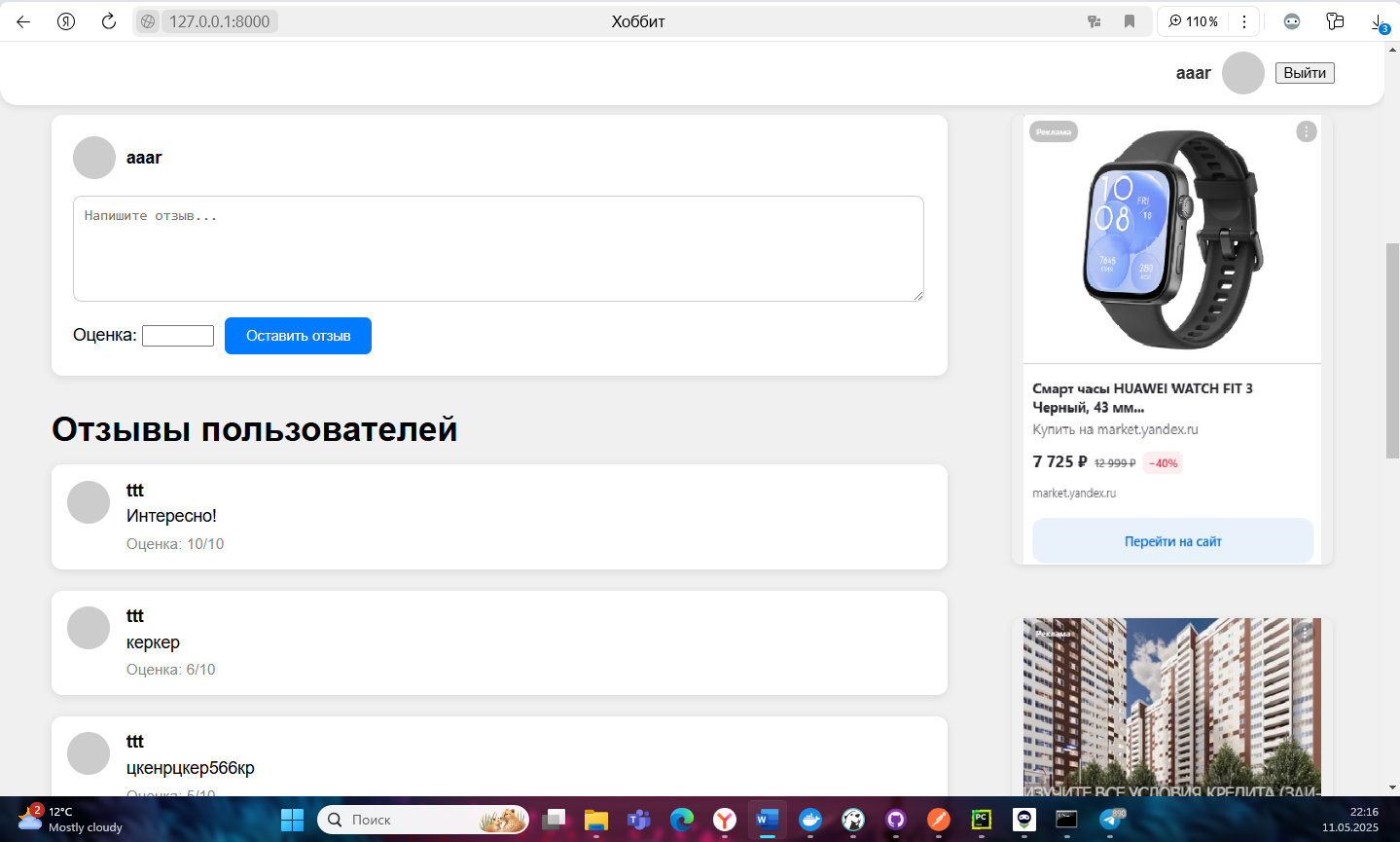
**3. Скриншоты страниц**











**4. Обработка данных и логика API**

**Регистрация и вход**

* При регистрации/авторизации данные отправляются через POST-запрос на сервер.
* В ответ сервер выдает JWT-токен.
* В views.py проверяется токен при каждом запросе, и пользователь определяется по нему.

**Пример кода обработки входа:**

def login\_view(request):

if request.method == 'POST':

username = request.POST['username']

password = request.POST['password']

response = requests.post(API\_URL + '/token/', data={'username': username, 'password': password})

if response.status\_code == 200:

token = response.json().get('access')

resp = redirect('/')

resp.set\_cookie('jwt\_token', token)

return resp

return render(request, 'login.html')

**Получение данных:**

* Данные фильмов, сериалов, отзывов получаются через requests.get(...) с заголовком Authorization: Bearer <token>.
* Пример:

response = requests.get(API\_URL + f'/films/{film\_id}/', headers={'Authorization': f'Bearer {token}'})

film = response.json()

return render(request, 'film\_detail.html', {'film': film})

**6. Вывод**

Все основные задачи лабораторной работы были выполнены:

* Созданы шаблоны на Django (Jinja2) для всех нужных страниц.
* Реализована регистрация и авторизация пользователей с использованием JWT.
* Данные загружаются через REST API и отображаются в интерфейсе.
* Реализовано отображение отзывов и проверка авторизации.
* Интерфейс протестирован: данные отображаются корректно, взаимодействие с сервером работает.