**Отчет по проекту "Онлайн-кинотеатр"**

**1. Архитектура проекта**

Проект представляет собой онлайн-кинотеатр, состоящий из:

* С помощью Django будет разработан пользовательский интерфейс и серверная часть приложения

**Взаимодействие компонентов**

1. Пользователь отправляет запрос с фронтенда к API (например, на просмотр списка фильмов).
2. Django обрабатывает запрос, взаимодействует с базой данных PostgreSQL и отправляет ответ в формате JSON.
3. Фронтенд получает ответ и отображает информацию пользователю.

**2. API (REST)**

**Основные эндпоинты**

**Пользователи (Users)**

* POST /api/users/register – регистрация нового пользователя.
* POST /api/users/login – авторизация.
* GET /api/users/me – информация о текущем пользователе.
* PUT /api/users/me – обновление данных пользователя.
* DELETE /api/users/me – удаление аккаунта.

**Фильмы (Movies)**

* GET /api/movies – список всех фильмов.
* GET /api/movies/{movieId} – информация о конкретном фильме.
* POST /api/movies – добавление нового фильма (для админов).
* PUT /api/movies/{movieId} – обновление информации о фильме (для админов).
* DELETE /api/movies/{movieId} – удаление фильма (для админов).

**Сериалы (Shows)**

* GET /api/shows – список всех сериалов.
* GET /api/shows/{showId} – информация о конкретном сериале.
* POST /api/shows – добавление нового сериала (для админов).
* PUT /api/shows/{showId} – обновление информации о сериале (для админов).
* DELETE /api/shows/{showId} – удаление сериала (для админов).

**Сезоны (Seasons)**

* GET /api/shows/{showId}/seasons – список сезонов для сериала.
* GET /api/shows/{showId}/seasons/{seasonId} – информация о конкретном сезоне.
* POST /api/shows/{showId}/seasons – добавление нового сезона (для админов).
* PUT /api/shows/{showId}/seasons/{seasonId} – обновление информации о сезоне (для админов).
* DELETE /api/shows/{showId}/seasons/{seasonId} – удаление сезона (для админов).

**Эпизоды (Episodes)**

* GET /api/shows/{showId}/episodes – список эпизодов для сериала.
* GET /api/shows/{showId}/episodes/{episodeId} – информация о конкретном эпизоде.
* POST /api/shows/{showId}/episodes – добавление нового эпизода (для админов).

**Жанры (Genres)**

* GET /api/genres/{genresId} – список жанров.
* POST /api/genres – добавление нового жанра (для админов).

**Поиск (Search)**

* GET /api/search?q=... – поиск фильмов/сериалов по названию, жанру, году.

**Отзывы/Рейтинги (Reviews/Ratings)**

* GET /api/movies/{movieId}/reviews – список отзывов для фильма.
* GET /api/shows/{showId}/reviews – список отзывов для сериала.
* POST /api/movies/{movieId}/reviews – добавление отзыва фильма.
* POST /api/shows/{showId}/reviews – добавление отзыва сериала.
* PUT /api/movies/{movieId}/reviews/{reviewId} – обновление отзыва фильма.
* PUT /api/shows/{showId}/reviews/{reviewId} – обновление отзыва сериала.
* DELETE /api/movies/{movieId}/reviews/{reviewId} – удаление отзыва фильма.
* DELETE /api/shows/{showId}/reviews/{reviewId} – удаление отзыва сериала.

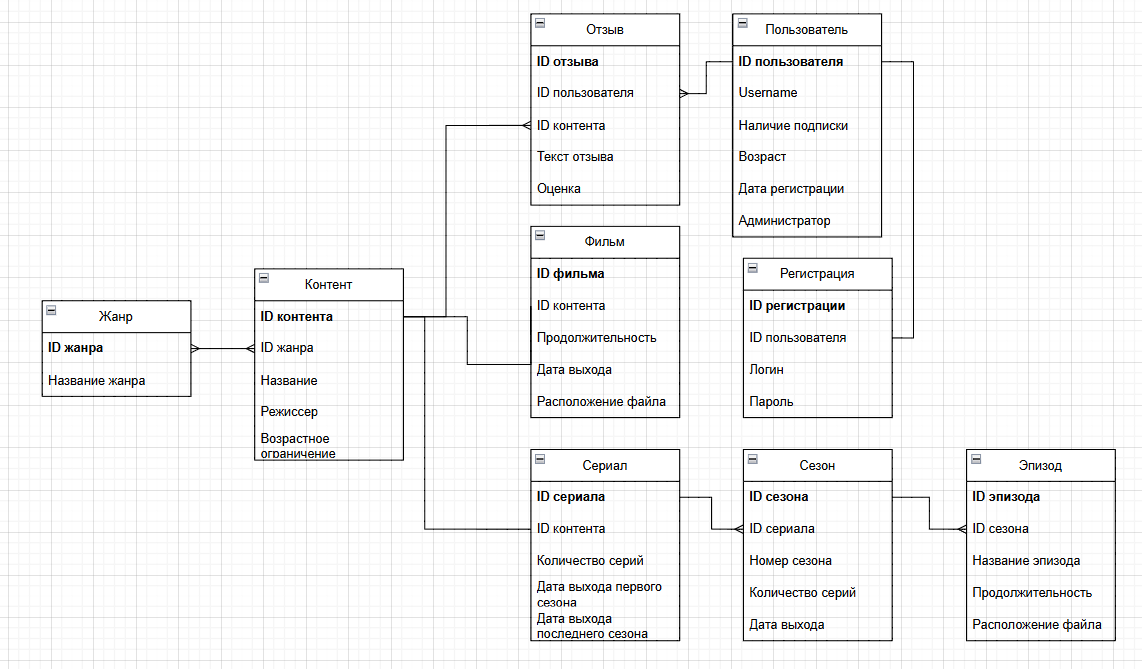
**Подписки и платежи (Subscriptions/Payments)**

* POST /api/users/me/subscriptions – покупка подписки.
* GET /api/users/me/subscriptions – информация о текущей подписке пользователя.

**3. Схема взаимодействия**

1. Пользователь регистрируется (POST /api/users/register) или входит в систему (POST /api/users/login).
2. Получает список фильмов (GET /api/movies).
3. Выбирает фильм, открывает его страницу (GET /api/movies/{movieId}).
4. Читает или оставляет отзыв (GET /api/movies/{movieId}/reviews, POST /api/movies/{movieId}/reviews).

**4. Схема базы данных**

****

База данных включает следующие сущности:

* Фильмы (ID, ID жанра, название, продолжительность, возрастное ограничение, дата выхода, режиссер, расположение файла)
* Сериалы (ID, ID жанра, название, количество серий, возрастное ограничение, дата выхода первого сезона, дата выхода последнего сезона, режиссер)
* Сезоны (ID, ID сериала, номер сезона, количество серий, дата выхода)
* Эпизоды (ID, ID сезона, название, продолжительность, расположение файла)
* Жанры (ID, название)
* Пользователи (ID, username, подписка, возраст, дата регистрации, статус администратора)
* Отзывы (ID, ID пользователя, ID фильма/сериала, текст, оценка)
* Регистрация (ID пользователя, логин, пароль)

**5. Стек технологий**

* **Фронтенд:** Django
* **Бэкенд:** Django
* **База данных:** PostgreSQL
* **Формат обмена данными:** JSON
* **Управление кодом:** Git, GitHub

**Дополнительно:** Docker и деплой пока не используются, но могут быть добавлены позже.