Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

федеральное государственное бюджетное образовательное учреждение высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

UI/UX

Лабораторная работа

Выполнил:

студент группы БВТ2002

Бартенев Т.Д.

**Задание**

Имеется API со списком фильмов, с которым нужно наладить взаимодействие.

Необходимо визуализировать данные полученные от API в виде таблицы, содержащей только важную (на ваш взгляд) информацию о фильме.

К каждому фильму из таблицы можно оставить комментарий (неограниченное количество), а также удалить его в любой момент времени, локальное состояние.

Ссылка на документацию API: <https://yts.mx/api>

Методы, с которыми необходимо работать:

* List Movies;
* Retrieve Movie.

Требования к задаче:

* NextJS;
* Отсутствие перезагрузки страницы;
* В приложении должна присутствовать пагинация.

Правила хорошего тона:

* Адаптивная верстка;
* Приятный и удобный UI;
* Простой и лаконичный код;

DRY.

# Теоретическая информация

На JavaScript выполняется большая часть интерактивных элементов на сайтах и в мобильных приложениях. JavaScript отлично работает с HTML/CSS и интегрирован основные браузеры на рынке. Чистый JavaScript используется в вебе, а для общего применения JavaScript разработчики используют различные среды выполнения, например, Node.js.

Node.js — это среда выполнения кода JavaScript. Она позволяет использовать JavaScript как язык программирования общего назначения: создавать на нем серверную часть и писать полноценные десктопные приложения.

Основа Node.js — движок V8. Этот движок был разработан Google и используется в браузере Google Chrome. Он компилирует код JavaScript в машинный код, который понимает процессор. Однако, чтобы сделать из JavaScript язык общего назначения, одного движка недостаточно. Так, например, для создания серверной части нужно, чтобы язык умел работать с файлами, сетью и т.п. Для решения этой проблемы разработчики добавили к V8 дополнительные возможности, с помощью своего кода и сторонних библиотек. В итоге у них получился инструмент, который превращает JavaScript в язык общего назначения.

Node.js стала популярна среди разработчиков благодаря возможности создавать серверную и клиентскую часть на одном языке, скорости работы и NPM.

**Выполнение**

Код представляет верстку с каталогом фильмов (Home), который включает в себя следующий функционал:

1. Стейт и эффекты:

- `useState` используется для хранения списка фильмов (`movies`), текущей страницы (`currentPage`), количества фильмов на странице (`moviesPerPage`), и выбранного жанра (`selectedGenre`).

- `useEffect` запрашивает список фильмов при загрузке страницы и обновляется при изменении `currentPage` или `selectedGenre`.

2. Загрузка списка фильмов:

- При помощи `fetch` происходит запрос к API YTS для получения списка фильмов с учетом текущей страницы и выбранного жанра.

- Результаты запроса сохраняются в состоянии `movies`.

3. Фильтрация по жанру:

- При клике на кнопку с жанром (например, 'Comedy', 'Drama', 'Animation' и т.д.), фильмы фильтруются по выбранному жанру при помощи `filterByGenre`.

- Если выбранный жанр совпадает с текущим `selectedGenre`, фильтр сбрасывается и отображаются все фильмы.

4. Отображение списка фильмов:

- Фильмы отображаются в блоке с классом `movie-grid`.

- Каждый фильм представлен как отдельный блок `movie-tile` с изображением обложки, рейтингом (`rating`) и названием (`title`) фильма.

- При клике на фильм происходит переход на страницу этого фильма (ссылка формируется с помощью `Link` из Next.js).

5. Пагинация:

- Кнопки пагинации внизу страницы отображают номера страниц для переключения между ними.

- При клике на кнопку с номером страницы вызывается функция `handlePagination`, которая обновляет `currentPage`.

6. Тема оформления:

- Используется контекст темы (`useTheme`), который позволяет изменять тему страницы между светлой и темной.

Этот компонент представляет основной контент главной страницы каталога фильмов, где пользователи могут просматривать список фильмов, фильтровать их по жанрам, а также переключаться между страницами с помощью пагинации.

Страница с детальной информацией о фильме (`MoviePage`). Что происходит в этом компоненте:

1. Запрос данных о фильме:

- Используется хук `useRouter` из Next.js для получения параметра `movieId` из URL.

- С помощью хука `useState` хранится состояние для информации о фильме (`movieData`), ключа страницы (`pageKey`), комментариев (`comments`), и текста комментария (`commentText`).

2. Загрузка данных о фильме:

- При помощи `useEffect` осуществляется запрос к API YTS с помощью `fetch`, чтобы получить информацию о фильме по `movieId`.

- Результат запроса сохраняется в состоянии `movieData`.

3. Комментарии:

- Используется массив `comments` для хранения комментариев к данному фильму. Он также сохраняется в локальное хранилище браузера.

- Функции `addComment` и `handleCommentTextChange` отвечают за добавление новых комментариев и обновление текста комментария соответственно.

4. Отображение информации о фильме:

- Если информация о фильме (`movieData`) еще не загружена, отображается текст "Loading...".

- Когда информация о фильме доступна, отображается основная информация о нем: название, обложка, длительность, дата загрузки, жанры, рейтинг, язык, ссылка на трейлер.

5. Отображение комментариев:

- Форма для добавления комментариев позволяет вводить текст комментария и отправлять его.

- Комментарии отображаются в списке, каждый комментарий представлен компонентом `CommentItem`, который включает текст комментария и кнопку для удаления.

6. Тема оформления:

- Применяется тема оформления (`dark-theme` или `light-theme`) в зависимости от выбранной темы.

7. Отображение обложки фильма:

- Используется тернарный оператор для проверки наличия обложки (`medium\_cover\_image`) фильма.

- Если обложка отсутствует, отображается блок с текстом "No cover available".

Этот компонент отображает информацию о конкретном фильме, включая его описание, комментарии пользователей и другие детали.

Главная страница со списком фильмов с темной темой (рисунок 1-2)

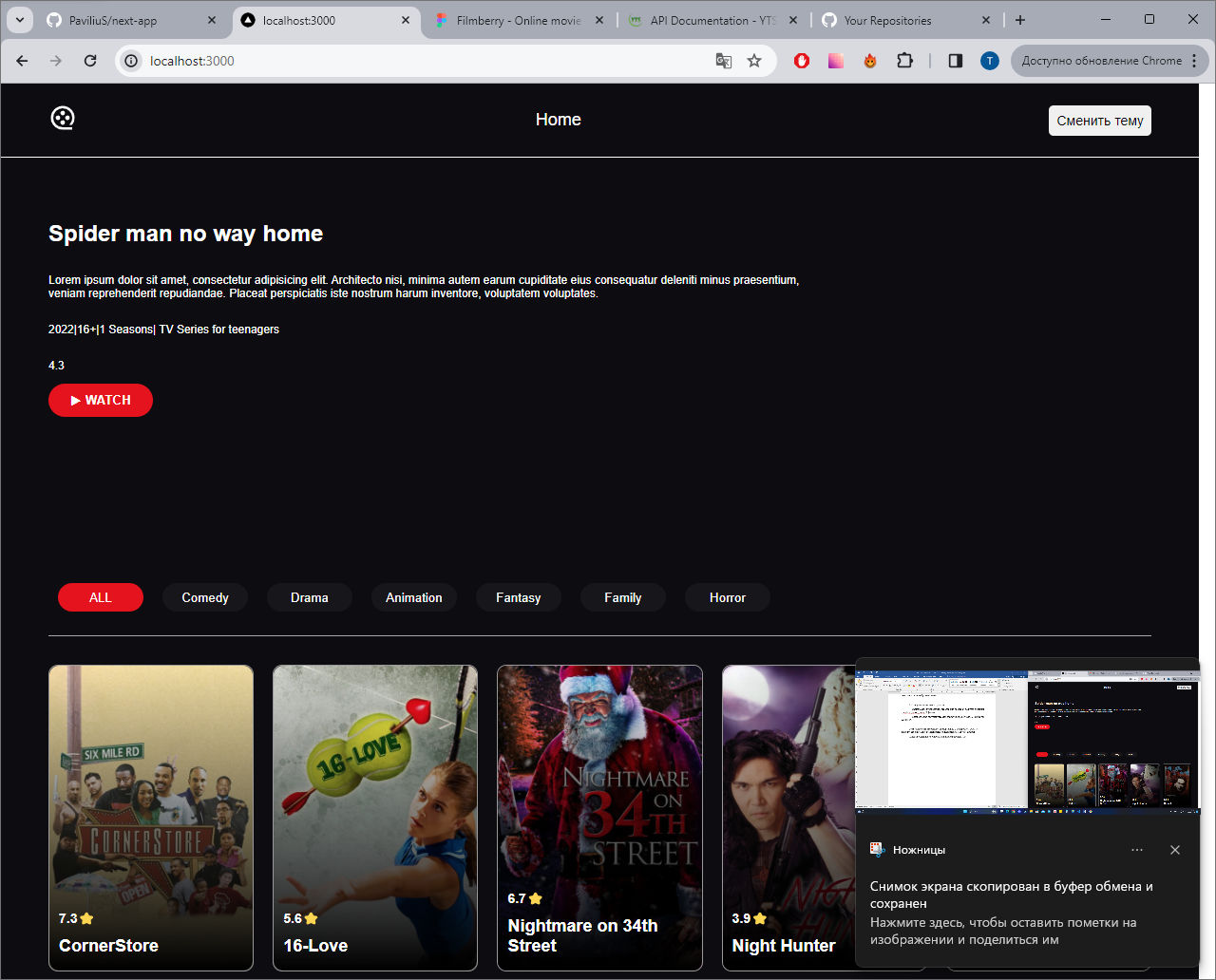


Рис. 1 – главная страница

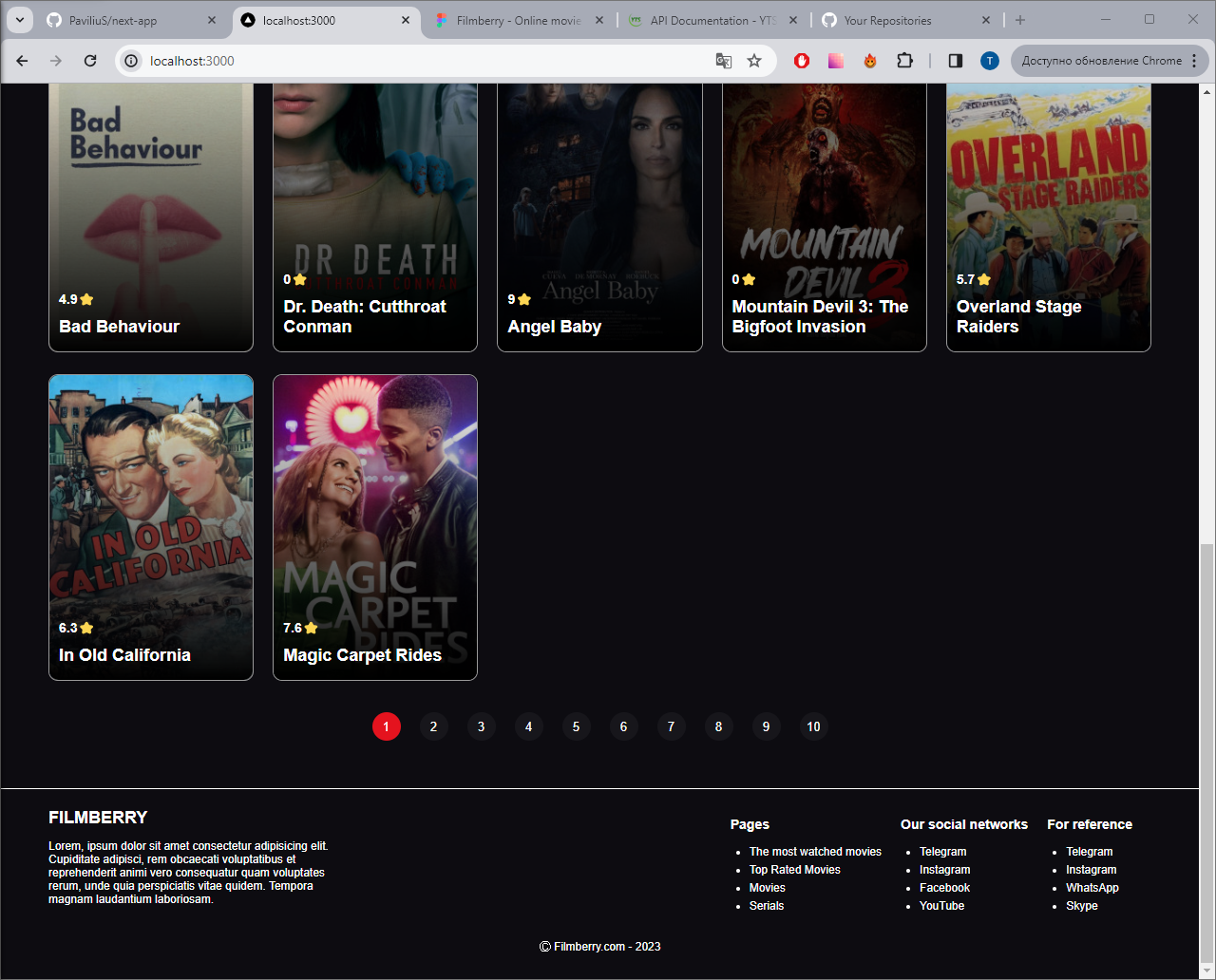


Рис. 2 - главная страница

Главная страница со светлой темой (рисунок 3)

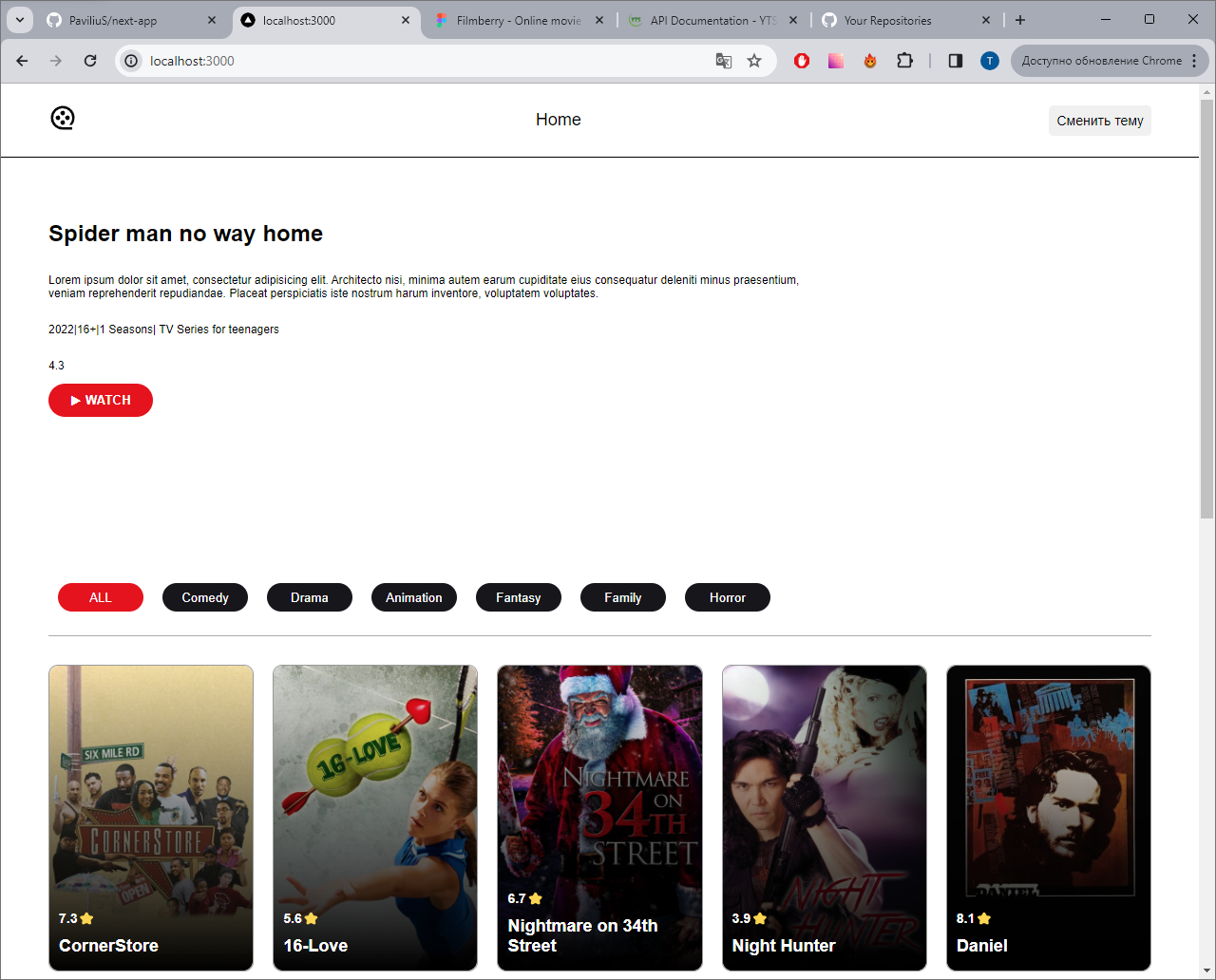


Рис. 3 - главная страница

Страница с детальной информацией с темной и светлой темой (рисунок 4-4.1)

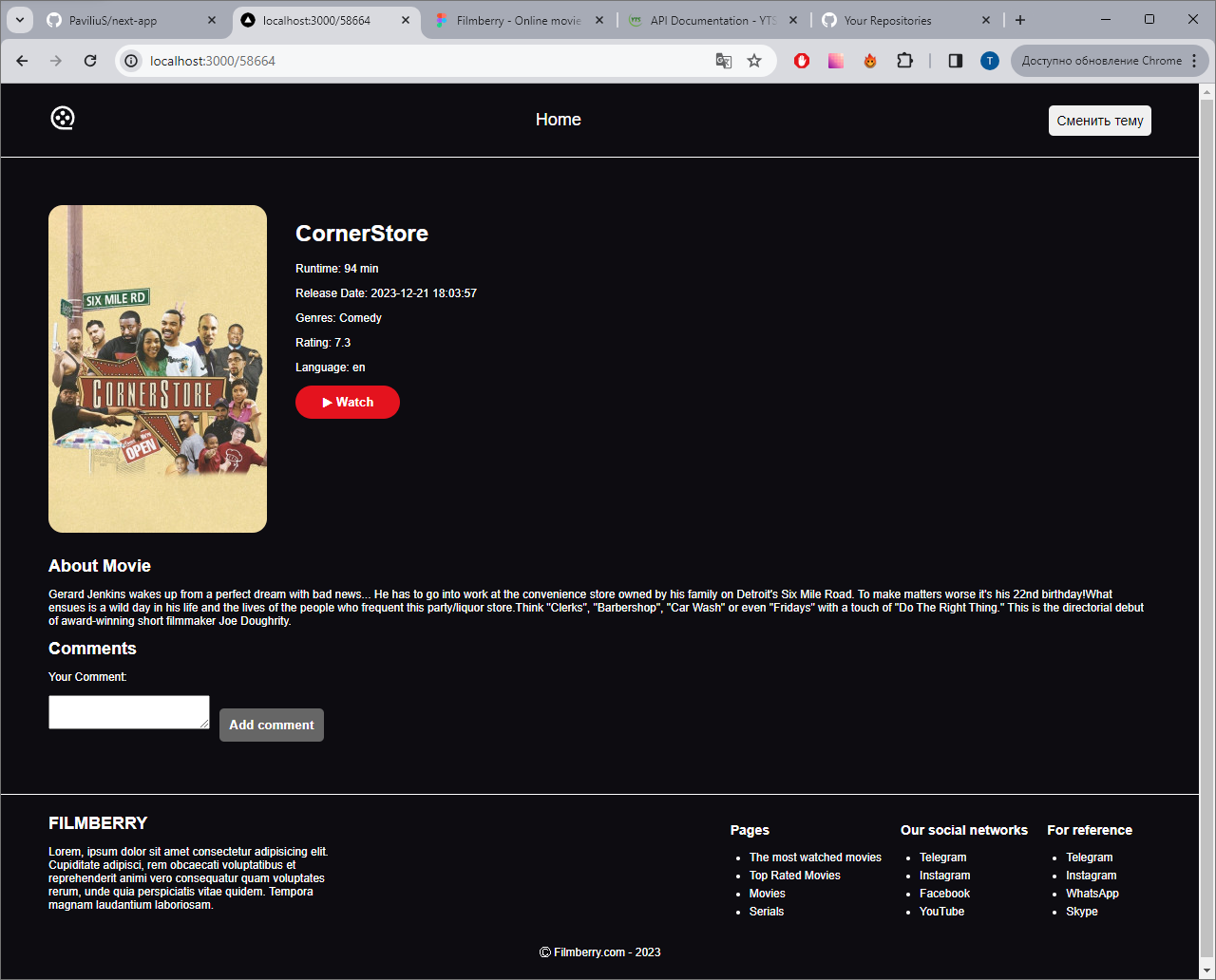


Рис. 4

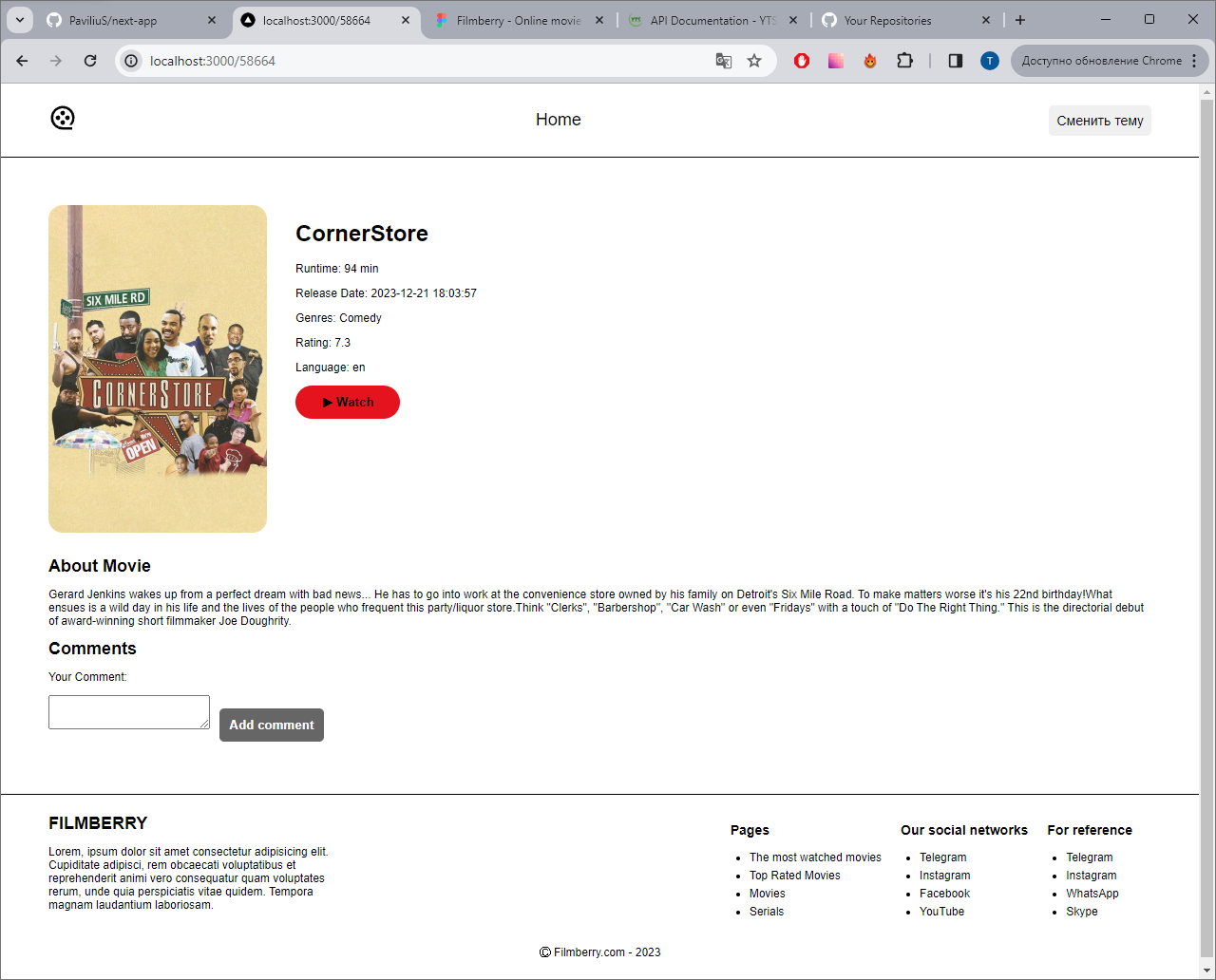


Рис. 4.1

Пример работы комментариев к фильму (рисунок 5)

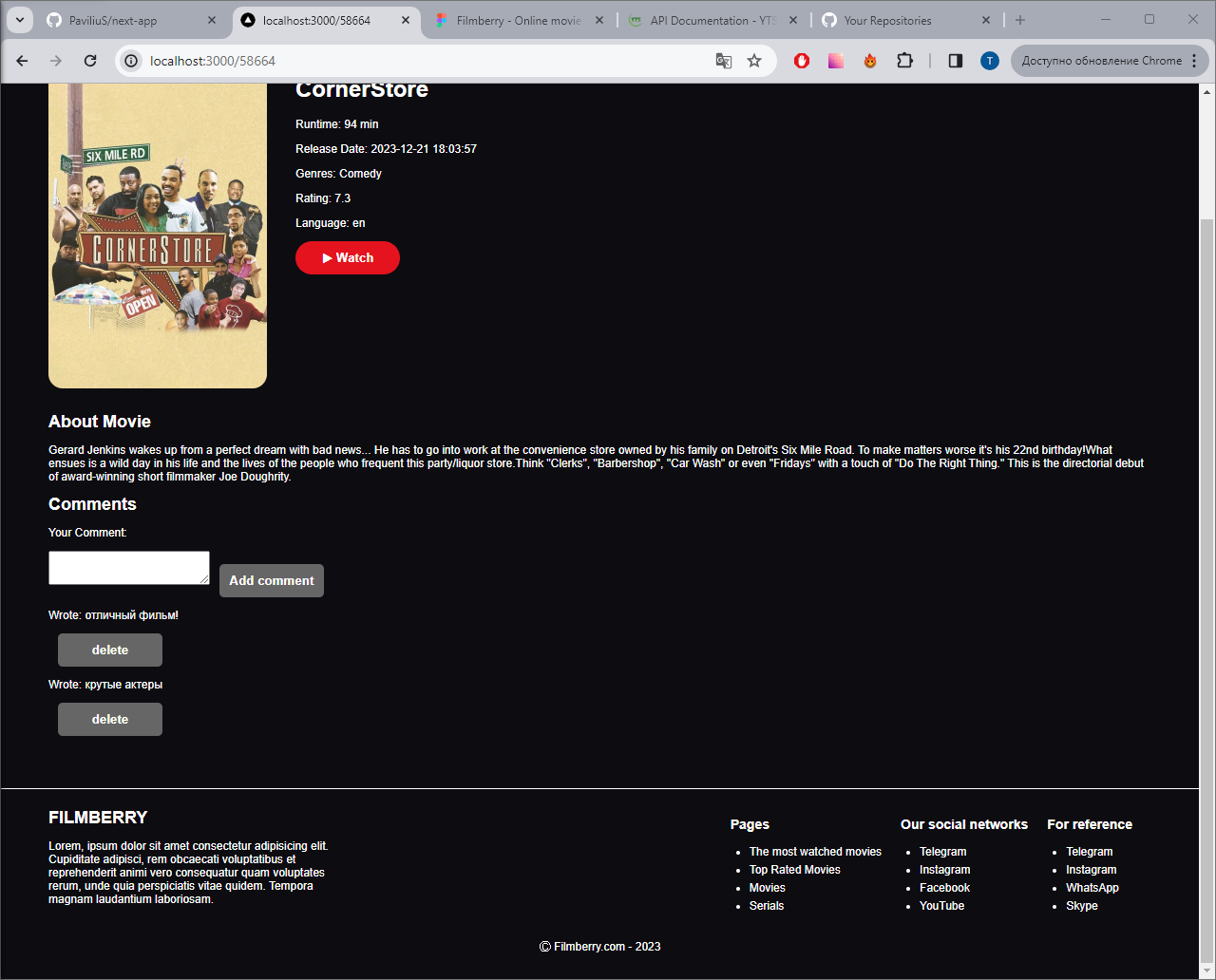


Рис. 5

**Вывод**

Данная лабораторная работа представляет верстку сайта-кинотеатра с детальной информации о фильмах, реализованную с использованием React и Next.js. Она включает в себя загрузку данных о фильме из внешнего API, отображение информации о фильме (название, обложка, рейтинг и прочее), возможность оставлять и удалять комментарии, а также использование локального хранилища браузера для сохранения комментариев. Этот проект показывает основные принципы работы с React для создания интерфейса детальной информации о фильмах с функциональностью взаимодействия пользователя.