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»

Отчёт по проекту

“Разработка сайта с фильмами”

 по дисциплине

“Основы UI и UX дизайна”

Выполнил:

студент группы БВТ2002

Захаров М. Ю.

Москва 2023 г.

Оглавление

[Оглавление 2](#_30j0zll)

Цель работы [3](#_1fob9te)

[Выполнение 3](#_3znysh7)

[Заключение 1](#_2et92p0)8

**Цель работы:**

Целью данной работы является изучение технологий NextJS и React Query, а также применение полученных знаний на практике с помощью настройки взаимодействия и корректного отображения данных, полученных из стороннего API (https://yts.mx/api), путем разработки сайта со списком фильмов. В рамках данного сайта необходимо реализовать отдельную страницу с подробным описанием и статистикой для каждого фильма, возможность добавления и удаления комментариев к фильмам, а также должна быть реализована адаптивная верстка страниц.

**Выполнение:**

Стоит начать с того, что для выполнения всего проекта в качестве языка программирования использовался TypeScript, а в качестве основной библиотеки используемых стилей, в соответствии с заданием, была взята Tailwind CSS.

Главная страница сайта состоит из Header`a и собственно списка с фильмами.

Header включает в себя Logo с соответствующей картинкой и заголовком сайта, а также кнопку смены светлой/темной темы на противоположную. В зависимости от выбранной темы меняются основные цвета фона и, для читаемости на данном фоне, отображаемого текста. Структура Header`a, а также его внешний вид отображения на сайте представлены на рисунках 1 и 2 соответственно.

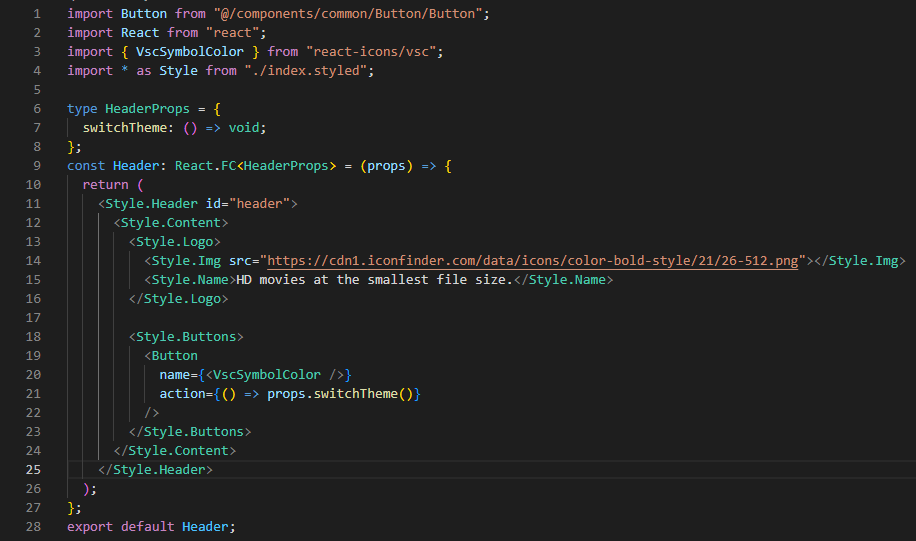


Рисунок 1 - Структура компоненты Header`a



Рисунок 2 - Внешний вид Header`a

Ниже под Header`ом на главной странице отображается сам список фильмов. Компонент Films, через который и реализован данный список, состоит из заголовка по центру экрана “Films” и собственно списка для отображения на одной странице filmlist, полученного в результате вызова хука useFilmList. Всего на одной странице отображается 24 фильма в 4 ряда по 6 фильмов в каждом. Общее количество (в данном случае 24) задается через параметр pageSize, который и указывается при вызове useFilmList. Структура реализации раздела со списком фильмов на языке TypeScript продемонстрирована на рисунке 3, а его внешний вид - на рисунке 4.



Рисунок 3 - Структура реализации раздела со списком фильмов

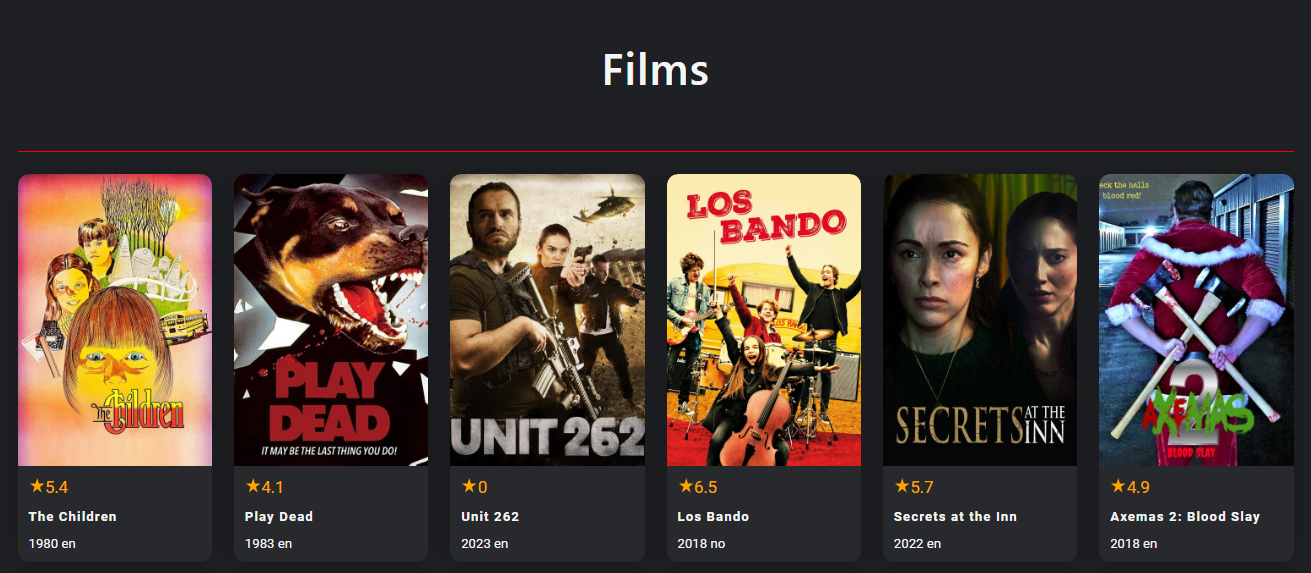


Рисунок 4 - Внешний вид раздела со списком фильмов

Список filmlist, используемый при отображении набора фильмов, представляет из себя, по сути, список, состоящий из компонентов Film, необходимых для отображения карточек конкретных фильмов, определяющихся по уникальному id.

Внутри компонента Film реализованы два компонента Card1 и Card2, предназначенные для разного отображения одной и той же карточки фильма при отсутствии наведения на нее курсора и при его наличии соответственно. Карточка фильма состоит из его обложки (либо специального фото-заглушки при ее отсутствии), а также расположенного под ней блока Content с определенной информацией о фильме: рейтинг, название, год выпуска и язык. При наведении курсора на карточку она выделяется на фоне остальных: блок Content подсвечивается, вокруг всей карточки образуется тень, фото фильма приближается и затемняется, а на его фоне отображаются жанры фильма (до 3 жанров) и специальная кнопка Details, при нажатии на которую открывается страница с подробным описанием данного фильма. Структура компонента Film представлена на рисунках 5 и 6, а внешний вид карточки фильма при наведении на нее курсора - на рисунке 7.

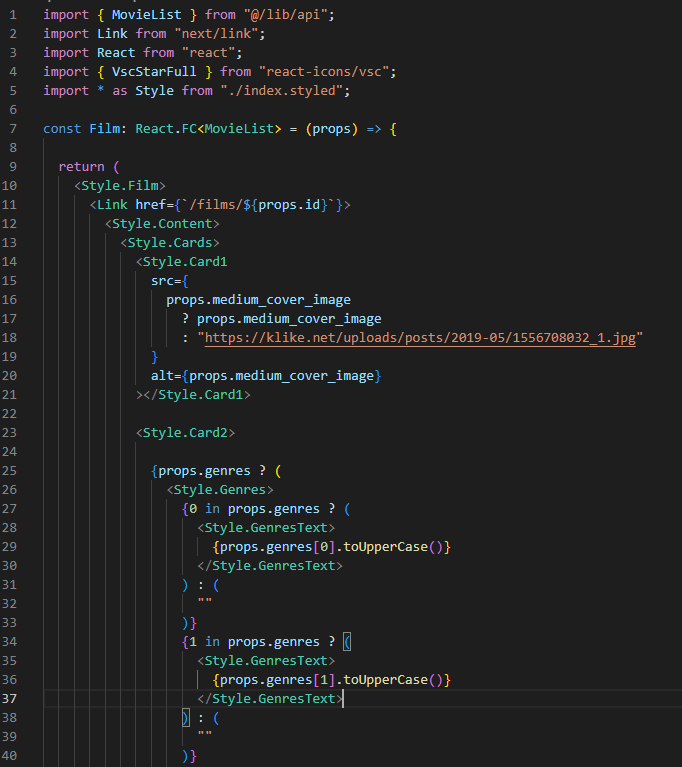


Рисунок 5 - Структура компонента Film (часть 1)



Рисунок 6 - Структура компонента Film (часть 2)

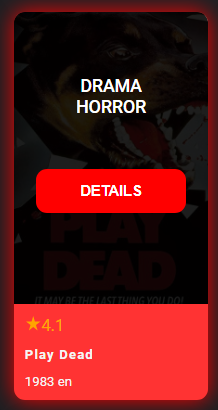


Рисунок 7 - Внешний вид карточки фильма при наведении на нее курсора

Также на сайте реализована работающая пагинация, внешний вид которой представлен на рисунке 8.

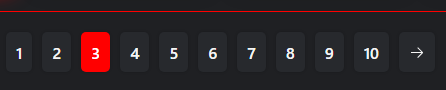


Рисунок 8 - Пагинация на сайте

Далее перейдем к странице с детальным отображением информации о фильме. На данной странице, под тем же самым Heder`ом, реализована внутри компонента ContentTitle кнопка “← Main page” для возвращения на главную страницу со списком фильмов. После идет компонент Data с информацией о выбранном фильме. Данная информация отображается по сути в два блока: в первом блоке располагается обложка фильма, год выпуска и язык, статистические данные (количество лайков и просмотров), жанры и кнопки для скачивания фильма в разном качестве, а во втором - название, рейтинг, описание, youtube-трейлер, и комментарии к фильму. Для CSS стилей компонента Data заданы свойства “display: grid” и “grid-template-columns: 1fr 2fr”. Таким образом, первый блок с инфомацией занимает треть пространства, а второй - оставшиеся две трети. Также в качестве фона используется расфокушенное изображение из фильма. Структура компонента Details с подробным описанием фильма показана на рисунках 9 и 10, а внешний вид данной страницы - на рисунке 11.

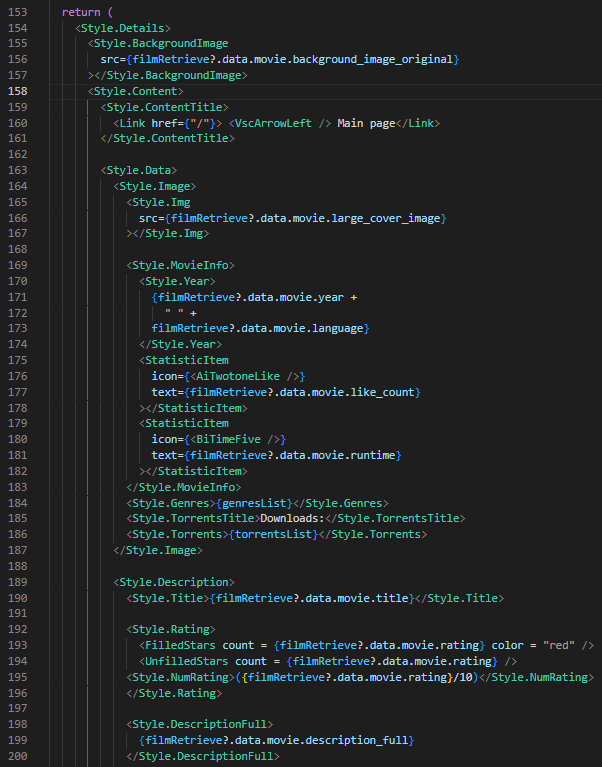


Рисунок 9 - Структура компонента Details с подробным описанием фильма (часть 1)

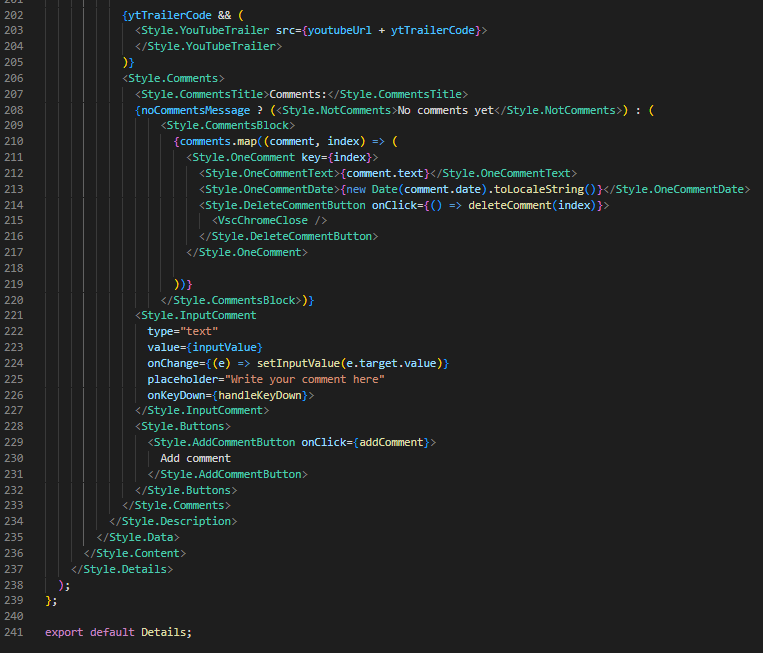


Рисунок 10 - Структура компонента Details с подробным описанием фильма (часть 2)

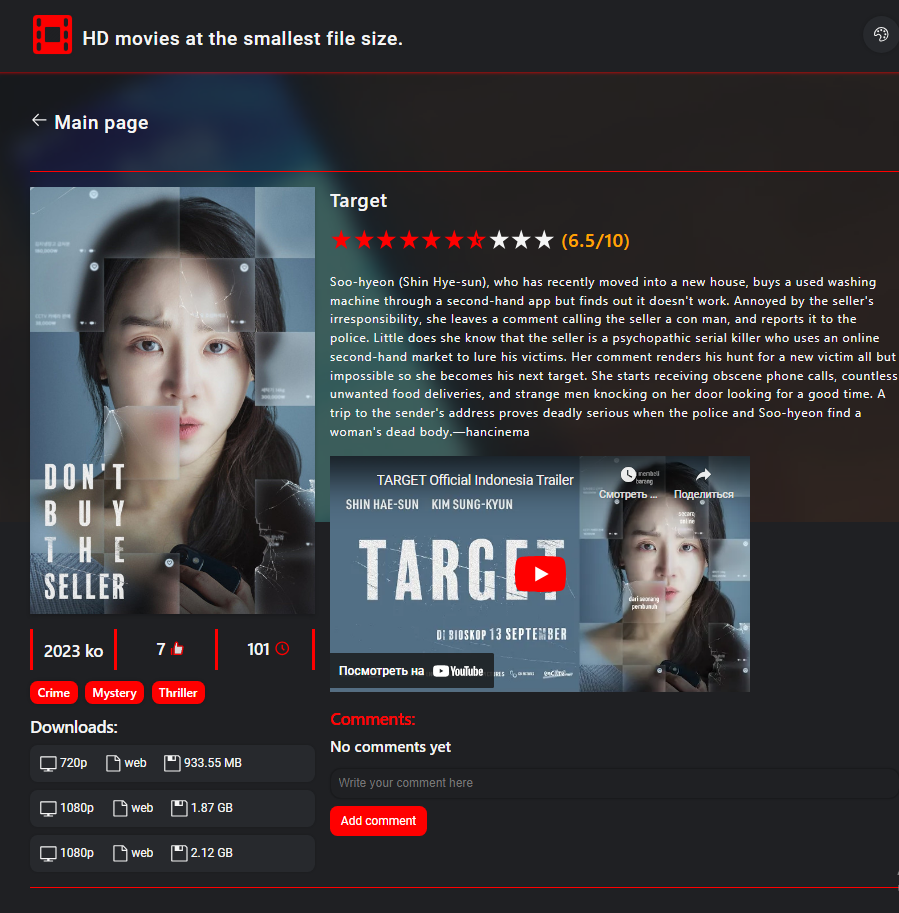


Рисунок 11 - Внешний вид страницы с подробным описанием фильма

Для отображения рейтинга фильма в виде десяти звезд были разработаны отдельно два компоненты FilledStars и UnfilledStars. Первый подсчитывает в соответствии с числовым значением рейтинга количество целиком/наполовину закрашенных изображений звезд VscStarFull/VscStarHalf и возвращает данной длины массив с изображениями, а второй - делает то же самое, но для оставшихся незакрашенных звезд. Структура компонента FilledStars представлена на рисунке 12, а компонента UnfilledStars - на рисунке 13 соответственно.

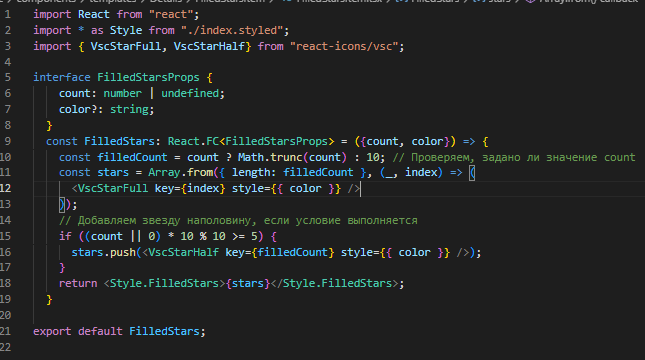


Рисунок 12 - Структура компонента FilledStars



Рисунок 13 - Структура компонента UnfilledStars

Как уже упоминалось ранее, на странице с детальной информацией о фильме реализована возможность оставления комментариев. Сами комментарии, вместе с точными датами их написания, хранятся в local storage и отображаются в строчку вместе с кнопкой их удаления. В случае если комментариев нет, выводится соответствующее сообщение. Для реализации данной логики комментирования фильмов были разработаны хук useEffect для считывания состояния: есть ли комментарии у конкретного фильма в данный момент или нет, а также методы addComment, deleteComment и handleKeyDown для добавления написанного комментария в local storage, удаления его оттуда и добавления комментария при нажатии кнопки Enter соответственно. Код реализации описанных методов представлен на рисунках 14 - 15, а внешний вид оставленного комментария на рисунке 16.

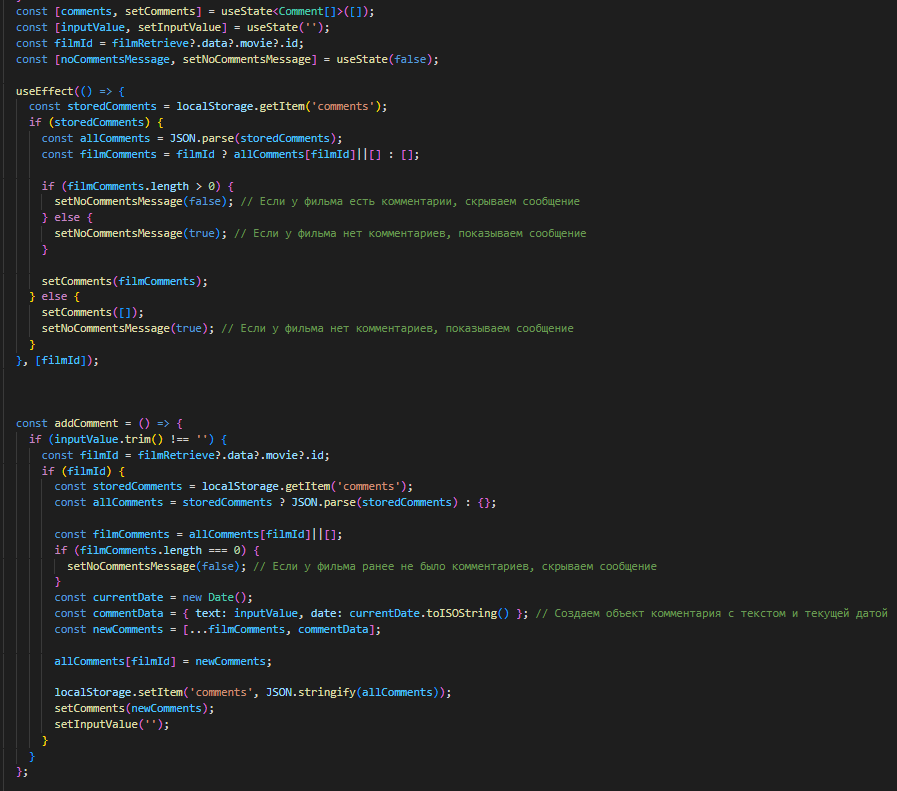


Рисунок 14 - Код реализации функционала комментирования (часть 1)

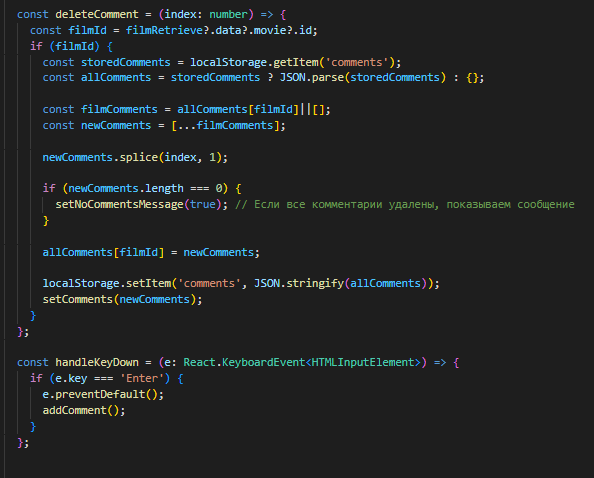


Рисунок 15 - Код реализации функционала комментирования (часть 1)

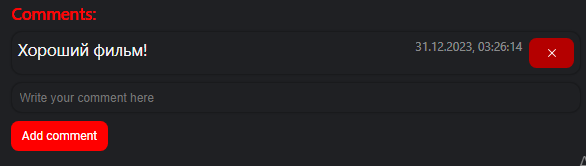


Рисунок 16 - Внешний фильм оставленного комментария к фильму

В конце выполнения работы мною была осуществлена работа по адаптивной верстке сайта для корректного отображения информации на различных разрешениях экрана. На главной странице с уменьшением разрешения уменьшается размер и количество в одном ряду карточек фильма, а на странице с подробным описанием фильма уменьшается шрифт текста и размер элементов. Примеры верстки главной страницы и страницы с деталями о фильме при разрешении экрана 375x667 (Iphone SE) продемонстрированы на рисунках 17 и 18 соответственно.

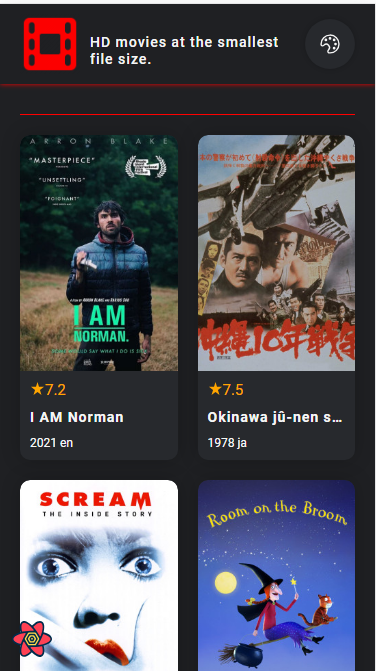


Рисунок 17 - Адаптивная верстка главной страницы при разрешении экрана 375x667 (Iphone SE)

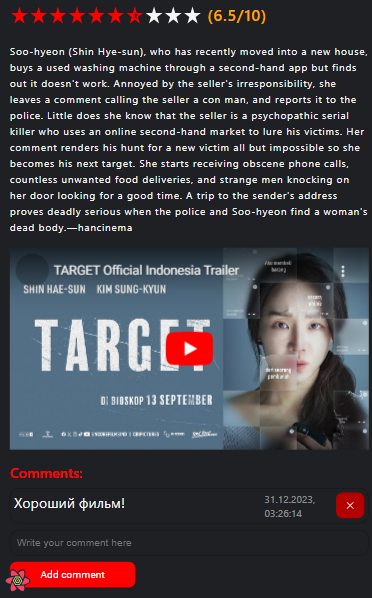


Рисунок 18 - Адаптивная верстка страницы с детальным описанием фильма при разрешении экрана 375x667 (Iphone SE)

Исходный код проекта хранится в github-репозитории по ссылке:

<https://github.com/MaximZakharov1/UIUX>

**Заключение:**

В ходе выполнения данной работы мною были изучены основы NextJS и React Query, а также реализован сайт, способный выводить на главной странице список фильмов в виде специальных карточек и подробную информацию о конкретном фильме на отдельной странице. Помимо этого, была проведена работа по адаптивной верстке сайта и были реализованы возможности скачивания torrent-файлов фильмов, а также добавления и удаления у них комментариев, хранящихся в local storage браузера. Со всеми поставленными задачами я справился, в результате чего приобрел незаменимые навыки frontend-разработки.