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»

Проект

по дисциплине «UI/UX»

Выполнил:

студент группы БВТ2001

Степанкова Я.В.

Москва 2023

**Введение**

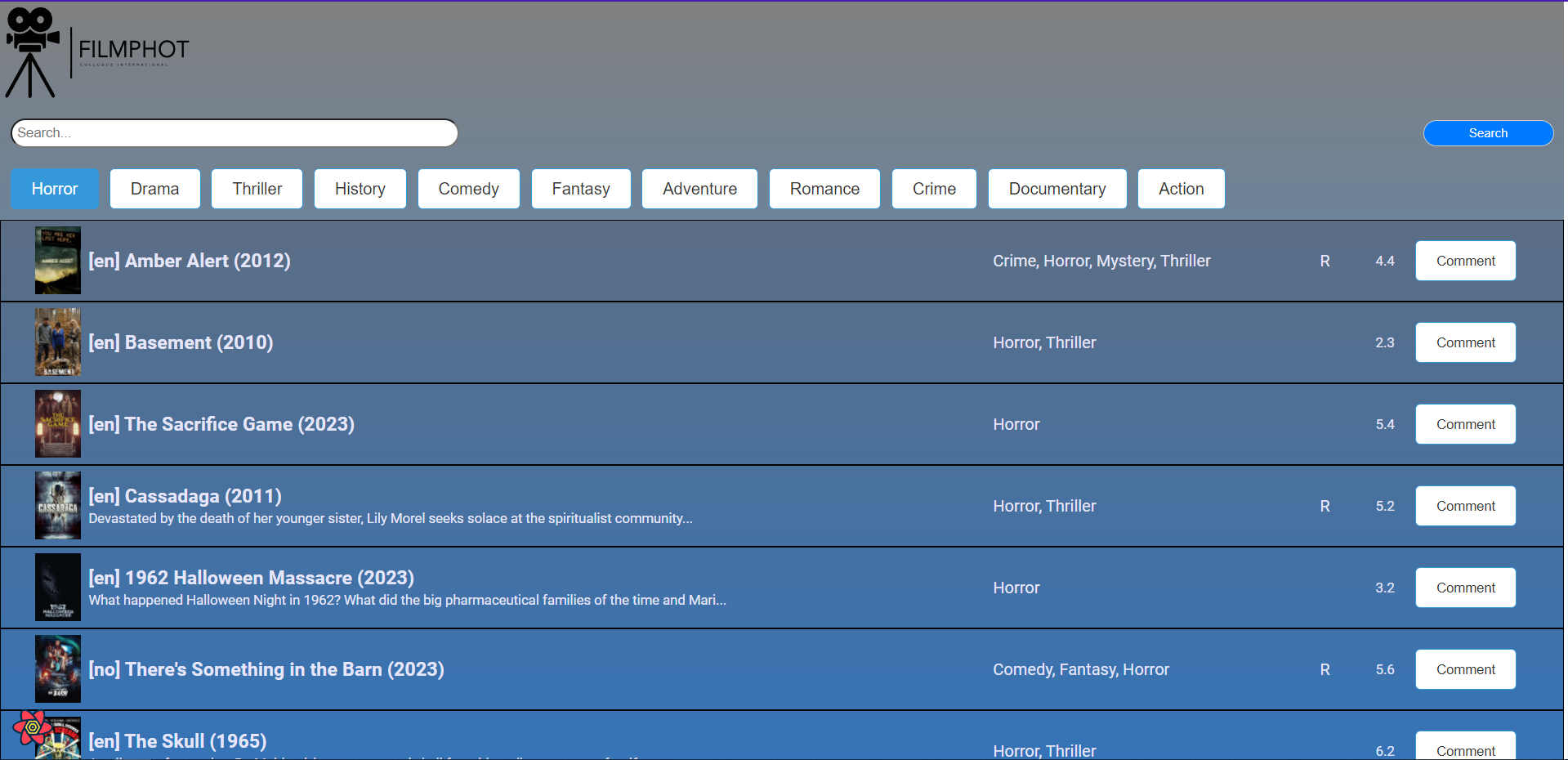
Проект представляет собой Next.js приложение, которое представляет собой сайт с выбором фильмов, похожее на Netflix.

Технологии и инструменты, используемые в проекте:

* Next.js
* React
* CSS
* TypeScript

**Выполнение**

Главную страницу я разделила на условные верхнюю и нижнюю части.



Скриншот 1 – главная страница

В верхней части находится лого сайта, поле поиска и кнопка поиска, категории фильмов. Категории фильмов – это кнопки, поэтому по ним таблица фильм отфильтровывается по выбранному жанру.

Нижняя часть страницы включает в себя таблицу с фильмами и пагинацию. Визуализация таблицы реализована с помощью хука useMovieList.ts. В нем используется Query Client провайдер. Также в хуке используется listMoviesQuery, в котором располагаются данные отображения фильмов, то есть та информация, которая будет отображаться на странице. Я экспортировала тип данных MovieItems. Этот тип включает в себя четыре поля: movie\_count, limit, page\_number и movies, где movies является массивом объектов типа ServerItem. В свою очередь в ServerItem находится полная информация о фильме, от id до данных для torrent скачивания. Эту информацию подробно я использую на станице с конкретным фильмом.

Отображение информации о фильме на главной странице и возможность оставить комментарий к фильму реализовано в файле MovieItem.tsx. Если подробнее, то:

* MovieItem Component: этот компонент принимает объект типа ServerItem в качестве своих свойств.
* Отображение информации о фильме: компонент отображает различные атрибуты фильма, такие как язык, название, год, жанры, рейтинг и т.д. Он также обрезает и отображает краткое описание фильма.
* Модальное окно для комментариев: при нажатии на кнопку "Comment" открывается модальное окно, которое позволяет пользователю оставить комментарий. После ввода комментария пользователь может отправить его, и модальное окно закроется. Комментарии сохраняются в localstorage.
* Использование хуков состояния: компонент использует хуки состояния, такие как useState, для управления отображением модального окна и хранения введенного комментария.
* Навигация: Компонент также использует компонент Link из библиотеки Next.js для навигации к странице с подробной информацией о фильме.

При нажатии на фильм есть возможность перейти на страницу с более детальным описанием фильма. Реализовано это в файле templates -> Details -> index.tsx.

* Details Component: этот компонент принимает идентификатор фильма в качестве своего свойства.
* Использование хуков состояния: компонент использует хук состояния useState, чтобы управлять отображением полного описания фильма.
* Использование хуков запросов: компонент использует хук запросов useFilmRetrieve для получения информации о фильме с сервера.
* Отображение информации о фильме: компонент отображает различные атрибуты фильма, такие как название, год, язык, жанры, рейтинг, количество лайков, продолжительность и т.д. Он также отображает краткое и полное описание фильма.
* Отображение загрузки: если информация о фильме еще не загружена, компонент отображает сообщение "Loading...".
* Реализована кнопка раскрытия и скрытия полного описания фильма – Read More \ Read Loss.
* Использование иконок: компонент использует иконки из библиотеки react-icons для отображения различных значков, таких как звезда, лайк, время и т.д.
* Отображение торрентов: компонент отображает список торрентов, доступных для загрузки.
* Навигация: компонент также использует компонент Link из библиотеки Next.js для навигации к странице с информацией о фильме.

Также есть возможность скачать фильм через torrent.

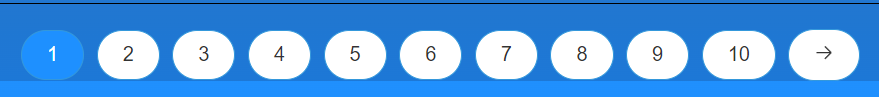
Пагинация:

Создается компонент пагинации (Pagination) для отображения списка страниц и управления ими. Компонент принимает несколько свойств, включая размер страницы (pageSize), текущую страницу (currentPage) и функцию обратного вызова onPageChange для изменения страницы.

Компонент рассчитывает общее количество страниц (pagesCount) и определяет размер порции (portionSize) в зависимости от ширины окна. Затем он формирует список страниц и отображает их в соответствии с текущей порцией. При клике на стрелки влево и вправо происходит изменение порции страниц.

Данный компонент позволяет удобно управлять пагинацией на веб-странице, а его поведение адаптируется к ширине окна браузера.

Также был изменен визуал пагинации.



Скриншот 2 – пагинация

Проект находится на GitHub по данный ссылке:

<https://github.com/KKopilka/my-app>