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»

Основы UI и UX дизайна

Лабораторная работа

Выполнил:

студент группы БВТ2001

Смирнов Д.А.

Проверил:

Дворянинов П.В.

Москва 2023

Оглавление

[Задание 3](#_Toc154070931)

[Выполнение 4](#_Toc154070932)

# 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.

# Выполнение

В самом начале я взялся за реализацию комментариев на странице подробностей о фильме. По пути components/templates/Details я создал новую папку CommmentItem, которая будет хранить все данные о том, как работают комментарии на любой странице. Дальше я создаю файл CommentItem.tsx, описывающий саму методику создания комментария. В нем также описывается функция для удаления комментария. В функцию создания комментария передаются параметры, представляющие из себя список уже существующих комментариев, а также событие, которое хранит в себе данные об идентификаторе, имени пользователя, текста комментария и функции по изменению состояния списка комментариев. Я также добавил стили к комментариям, создав соответствующий файл index.styled.ts. Помимо этого в нем я добавил способы отображения имени пользователя, текста комментария и кнопки для удаления. Файлы, описывающие логику работы комментариев и их стиля, показаны на рисунках 1, 2 и 3.

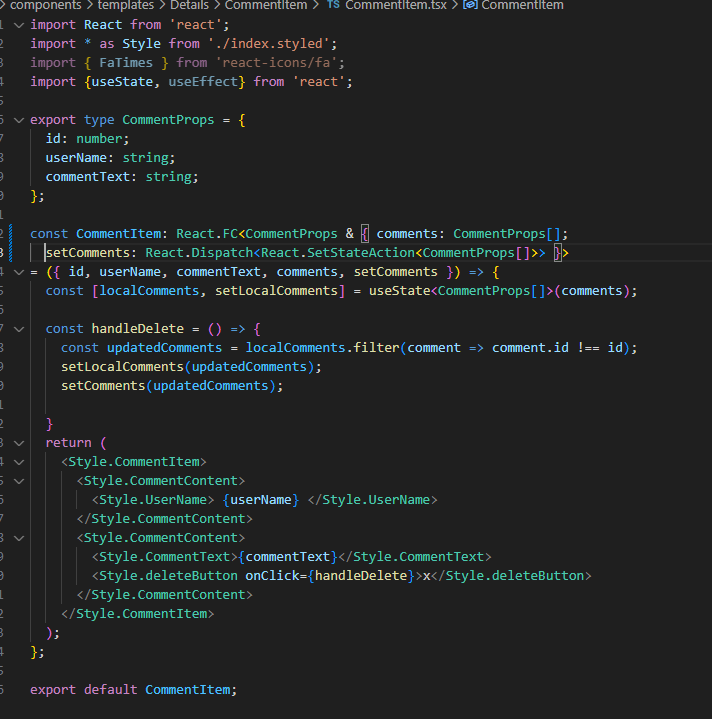


Рисунок 1 – Файл с логикой работы комментариев

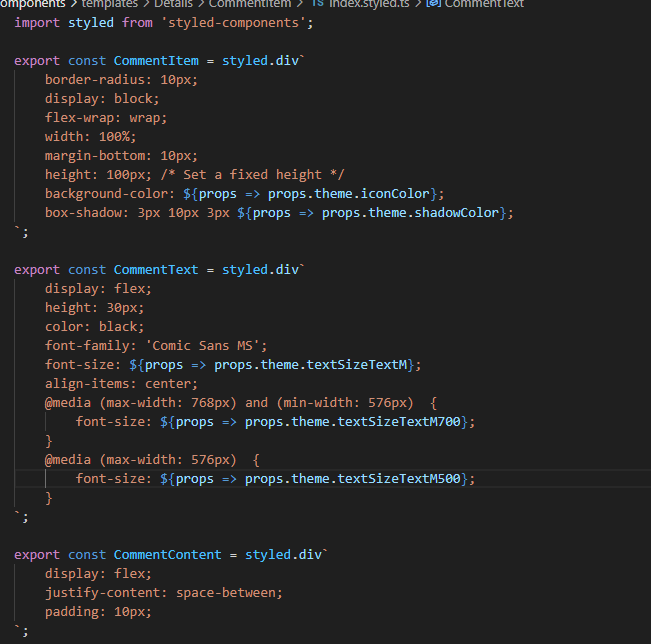


Рисунок 2 – Первая часть файла стилей комментариев



Рисунок 3 – Вторая часть файла стилей комментариев

Затем идет процедура добавления форм для принятия деталей комментария, а именно имени пользователя и текста комментария. Для того, чтобы это осуществить, я в файле index.tsx в функцию, описывающей работу страницы с подробностями, добавляю поле pageKey для url-страницы, чтобы для каждого фильма отображались только его комментарии. Затем в это поле с помощью вызова window.location.href передаю url-страницы. Считывания url-страницы представлено на рисунке 4.

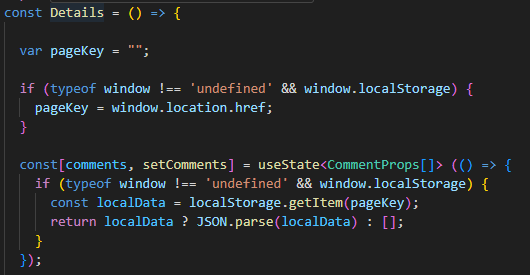


Рисунок 4 – Считывание url-страницы

Следующим шагом является добавление функции для отображения списка комментариев на странице. Данный список записывается в localStorage, из которого будет происходить его считывание. При его отсутствии будет выводится пустота. Также с помощью хука описываю запись комментария в localStorage через функцию setItem. Запись комментария через хук представлена на рисунке 5.

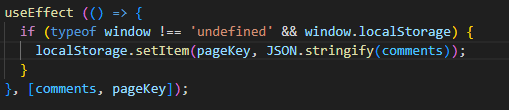


Рисунок 5 – Запись комментария через хук

После этого я описываю функцию для добавления комментария addCommnent, принимающая на вход событие, из которого затем достаются данные о комментарии. Функция для добавления комментария представлена на рисунке 6.

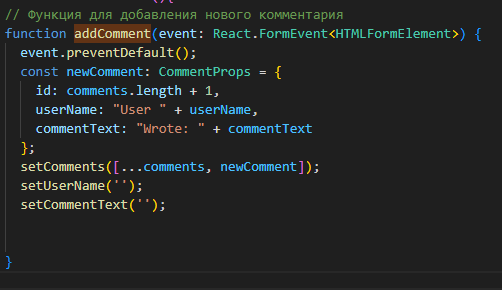


Рисунок 6 – Функция добавления комментария

Далее я описываю обработчики изменения полей ввода имени пользователя и текста комментария. Данные обработчики представлены на рисунке 7.

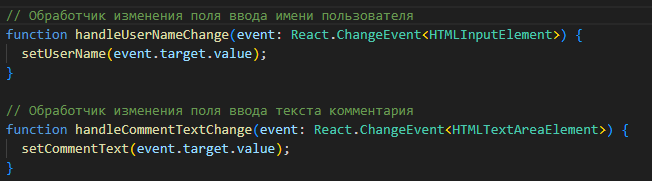


Рисунок 7 – Обработчики ввода имени пользователя и текста комментария

Потом создаю массив элементов React, каждый из которых представляет собой отдельный комментарий. Также добавляю на страницу поля формы для ввода имени пользователя и текста комментария и кнопку для отправки комментария в localStorage. Данные действия представлены на рисунке 8.

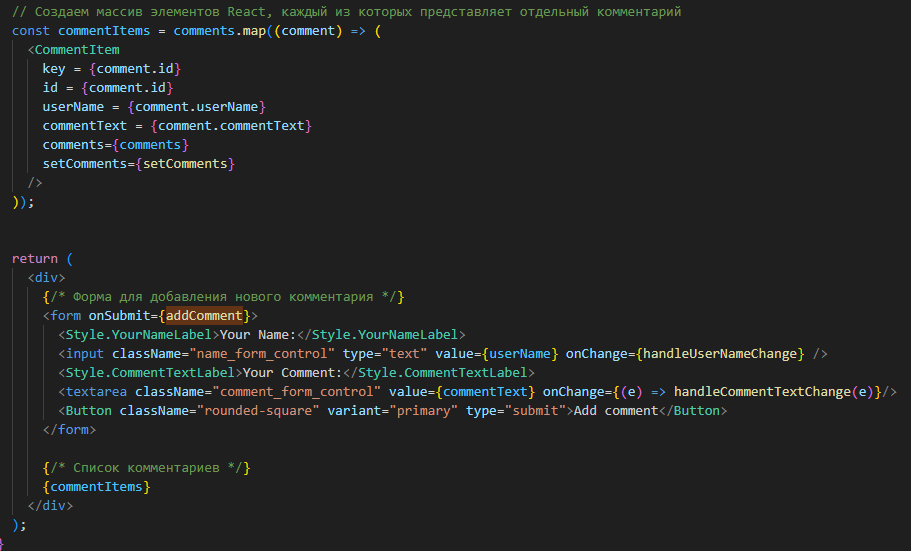


Рисунок 8 – Создание массива элементов React и добавление на страницу полей формы для отправки комментариев

Ниже представлен рисунок, на котором показана страница подробностей фильма с добавленными полями формы комментариев.



Рисунок 9 – Страница подробностей фильма с полями для отправки комментариев

Так выглядит страница с добавленными комментариями

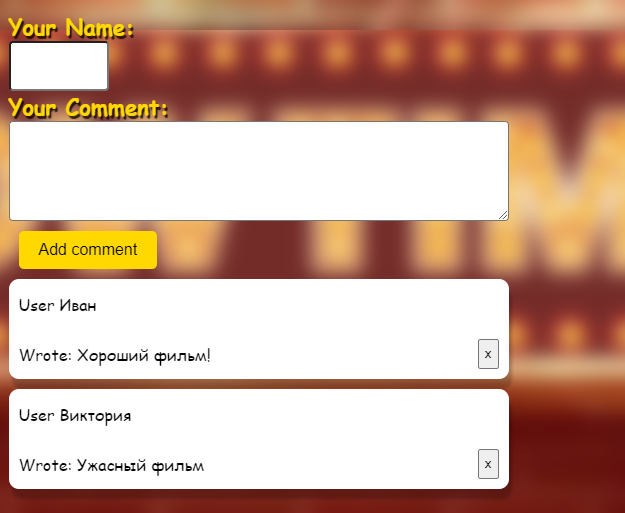


Рисунок 10 – Страница с добавленными комментариями

Для изменения шрифтов темы и изображения бэкграунда я изменял файл theme.ts, который отвечает за общую стилизацию сайта. Также я добавил в качестве эффектов размытие картинки на фоне и фиксацию изображения на одном месте. Код, отвечающий за это, представлен на рисунке 11.

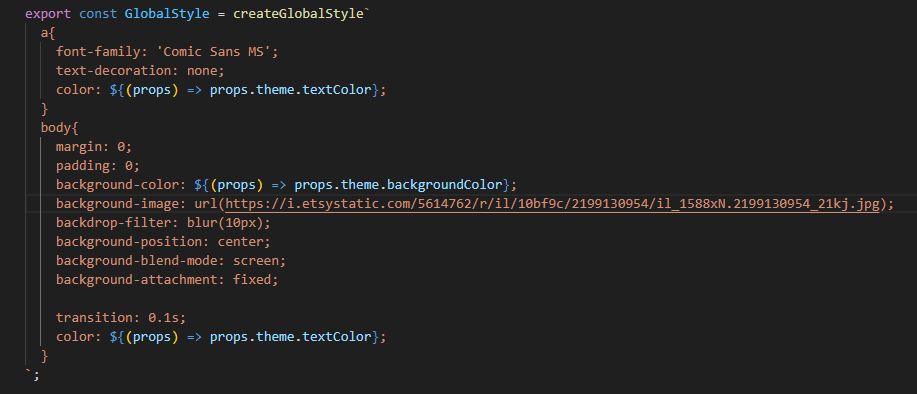


Рисунок 11 – Изменение шрифтов темы и добавление картинки бэкграунда с эффектами

Также я изменил цвета темы, расположенные в том же файле. Цвета темы представлены на рисунке 12.



Рисунок 12 – Изменение цветов темы

Следующим шагом было добавление адаптивной верстки к сайту. Это достигалось с помощью хука useEffect, который проверял значения поля windowWidth и на основе этого менял кол-во карточек фильмов на экране. Код адаптивной верстки представлен на рисунке 13.

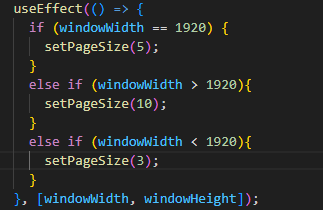


Рисунок 13 – Адаптивная верстка

Далее будет показана работа адаптивной верстки. Она представлена на рисунках 14-16.



Рисунок 14 – Адаптивная верстка при ширине экрана в 1920 пикселей



Рисунок 15 – Адаптивная верстка при ширине экрана больше 1920 пикселей

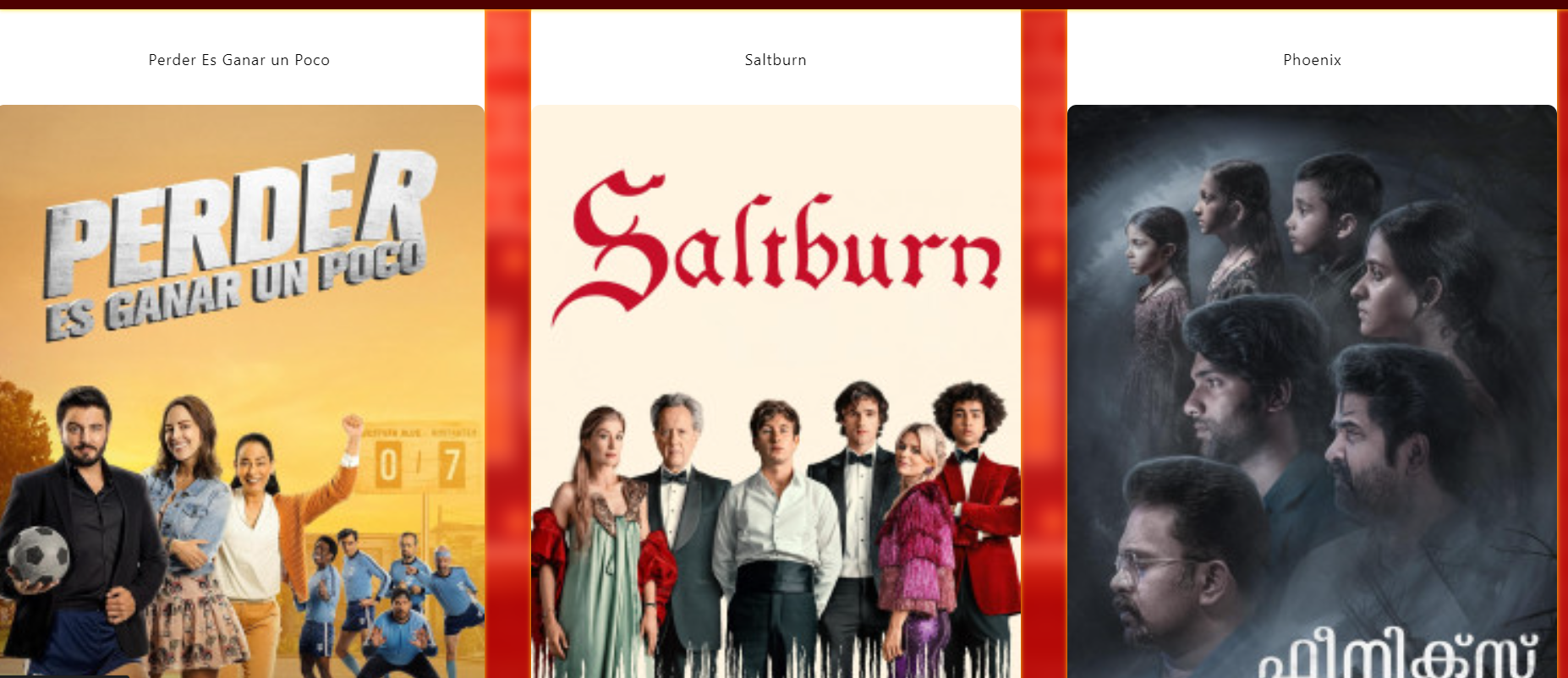


Рисунок 16 – Адаптивная верстка при ширине экрана меньше 1920 пикселей

Для пагинации также делаю адаптивную верстку. При различных значениях ширины экрана меняется стиль кнопок. Код данной верстки представлен на рисунке 17.

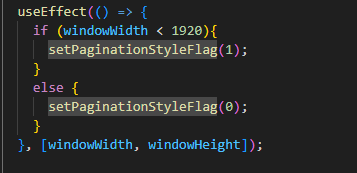


Рисунок 17 – Код адаптивной верстки для пагинации

На рисунках 18-20 представлены стили пагинации для адаптивной верстки.

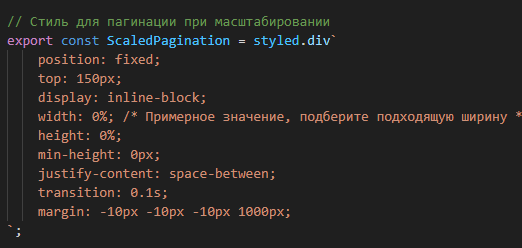


Рисунок 18

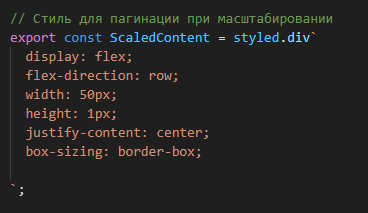


Рисунок 19

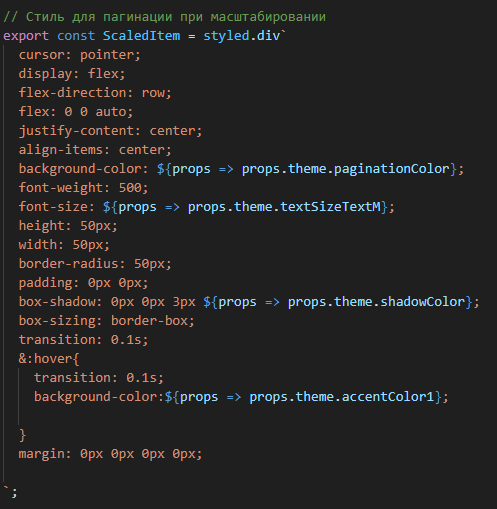


Рисунок 20

Настройки стилей карточек фильмов представлены в файле index.styled.ts по пути templates/Films/Film. На рисунках 21-28 представлены параметры стилей карточек фильмов.



Рисунок 21

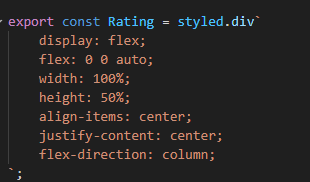


Рисунок 22

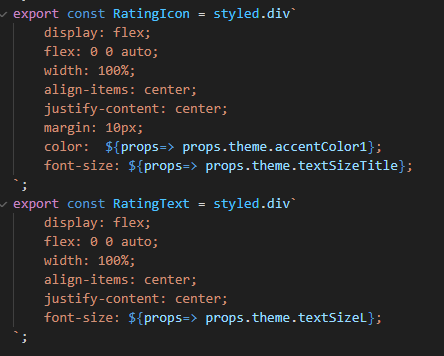


Рисунок 23

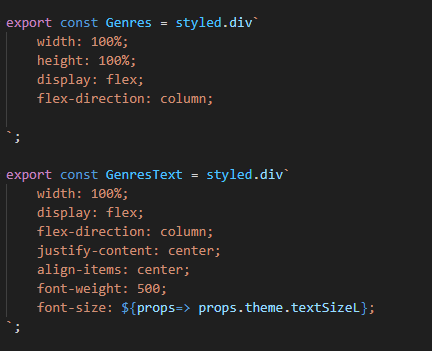


Рисунок 24

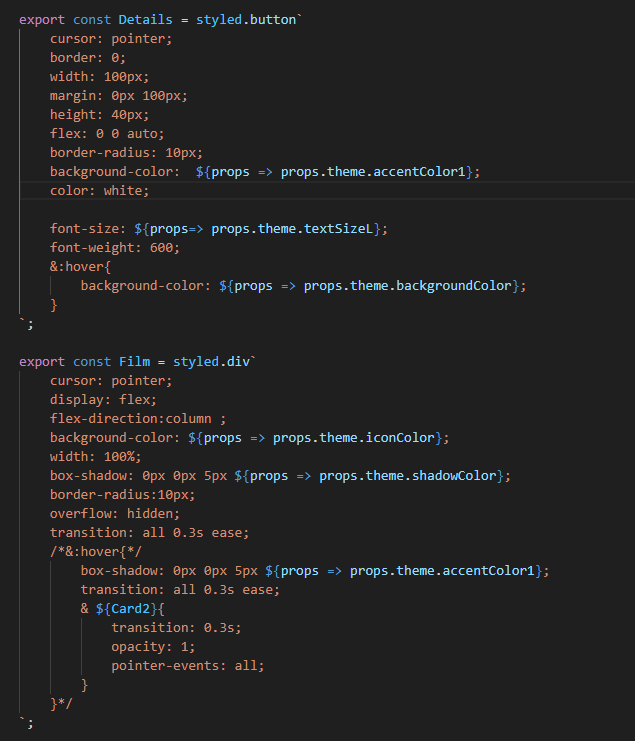


Рисунок 25

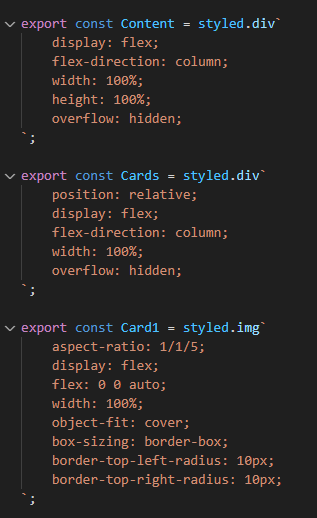


Рисунок 26

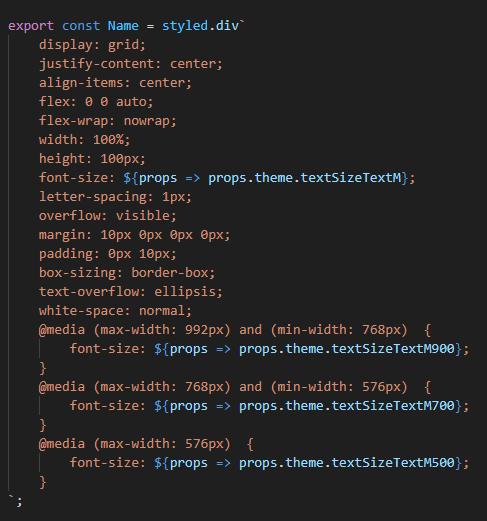


Рисунок 27

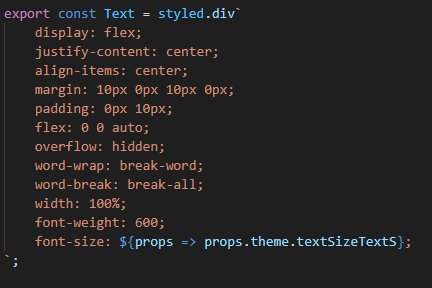


Рисунок 28

Далее на рисунке 29 будет показан вид карточек фильмов, настроенный в файле стилей.

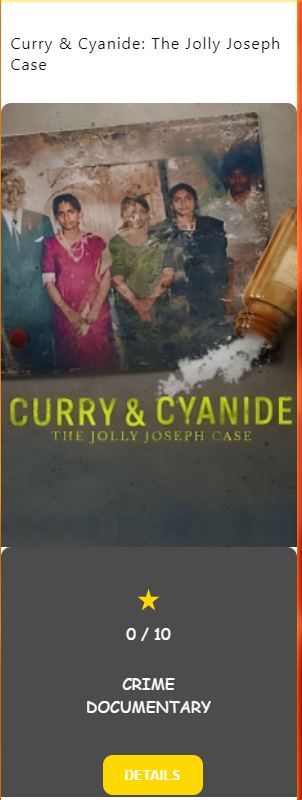


Рисунок 29 – Внешний вид карточки фильма

# Вывод

В ходе данной лабораторной работы я изучил синтаксис языка TypeScript, сделал сайт для фильмов, используя API axios, и добавил свои стили к его элементам. Также была реализована адаптивная верстка сайта и реализована возможность оставлять комментарии к каждому фильму.