Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Отчёт по проектной работе

по дисциплине

«UI/UX»

Выполнил:

студент группы БВТ2003

Аджиумеров Александр Арсенович

Проверил:  
Дворянинов Павел Владимирович

Москва, 2023

Оглавление

**[Цель работы 3](#_Toc19016)**

**[Результаты выполненной работы 4](#_Toc20197)**

**[Структура проекта 14](#_Toc15729)**

**[Листинги кода 16](#_Toc21911)**

**[Раздел комментариев 16](#_Toc5875)**

**[Карточки фильмов на главной странице 20](#_Toc21992)**

**[Рейтинг на странице сайта 25](#_Toc28319)**

**[Хэдер 26](#_Toc18337)**

**[Статистика на странице фильма 28](#_Toc10070)**

**[Страница фильма 29](#_Toc2938)**

**[Вывод 37](#_Toc19194)**

# Цель работы

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.

# Результаты выполненной работы

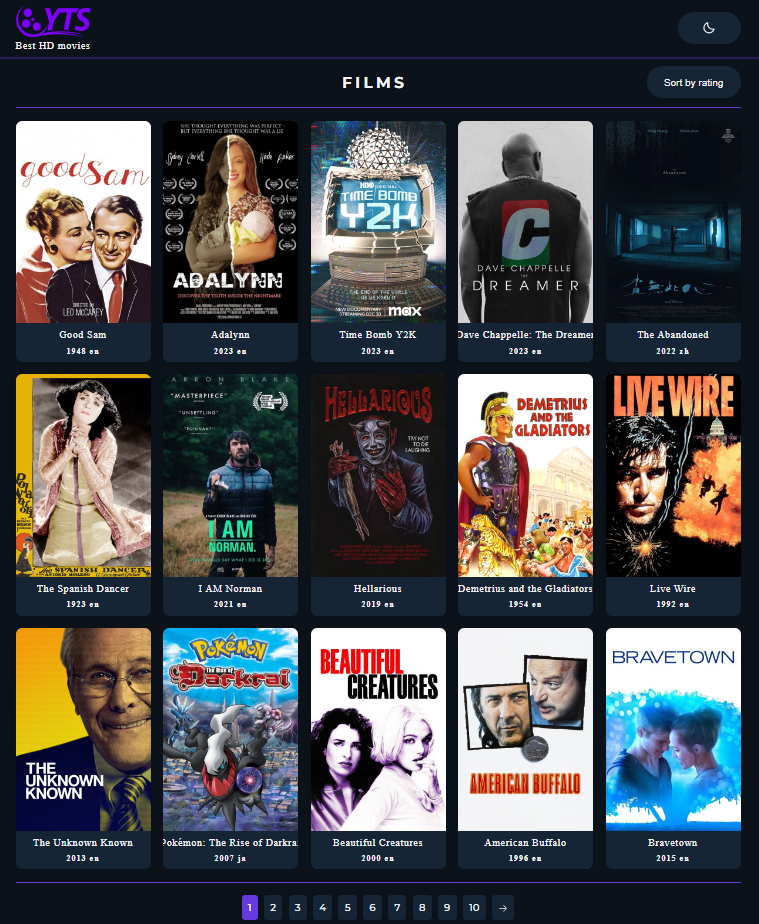


Рисунок 1 - главная страница (тёмная тема)

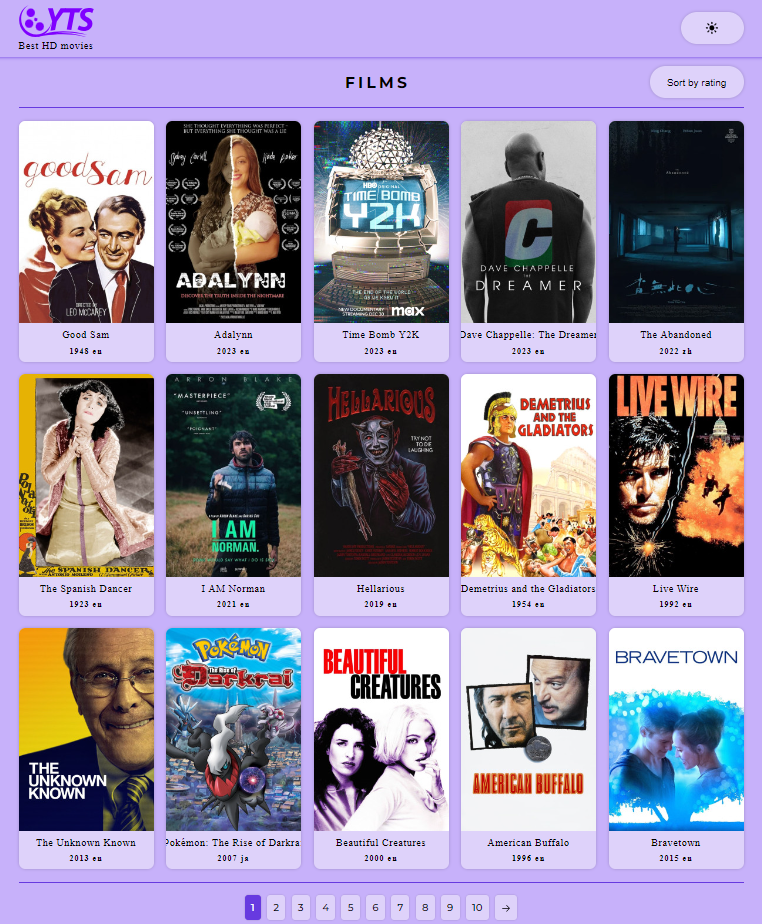


Рисунок 2 - главная страница (светлая тема)

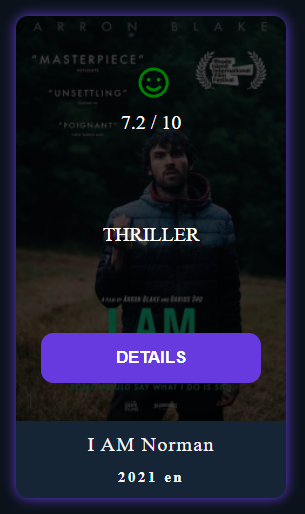


Рисунок 3 - курсор на фильме (тёмная тема)

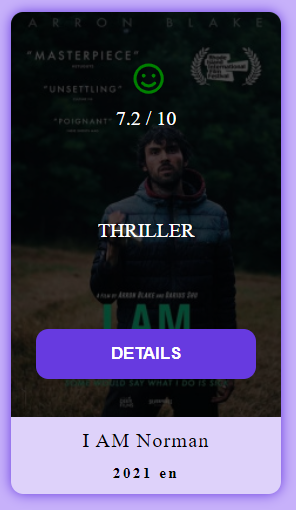


Рисунок 4 - курсор на фильме (светлая тема)



Рисунок 5 - страница фильма (тёмная тема)



Рисунок 6 - страница фильма (светлая тема)

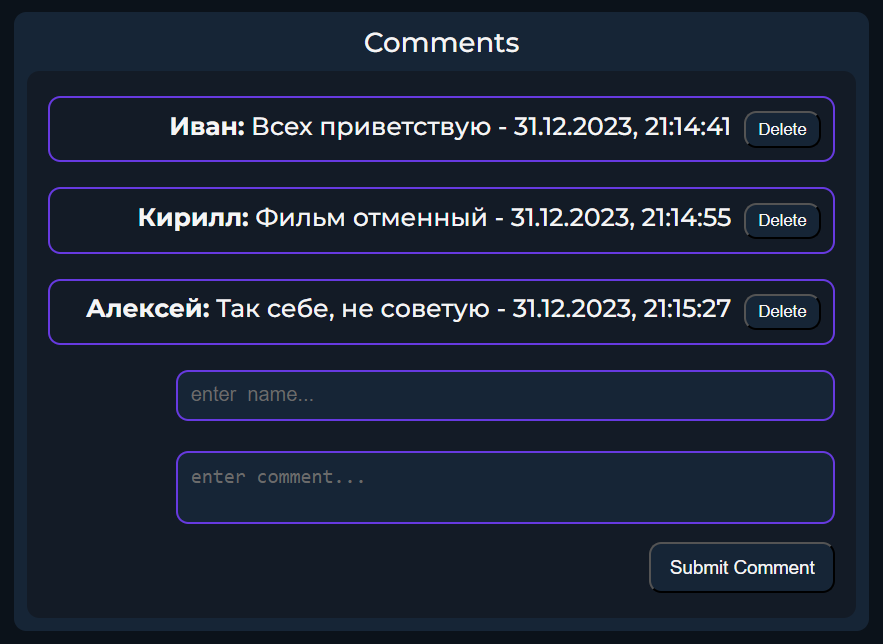


Рисунок 7 - отображение комментариев (тёмная тема)

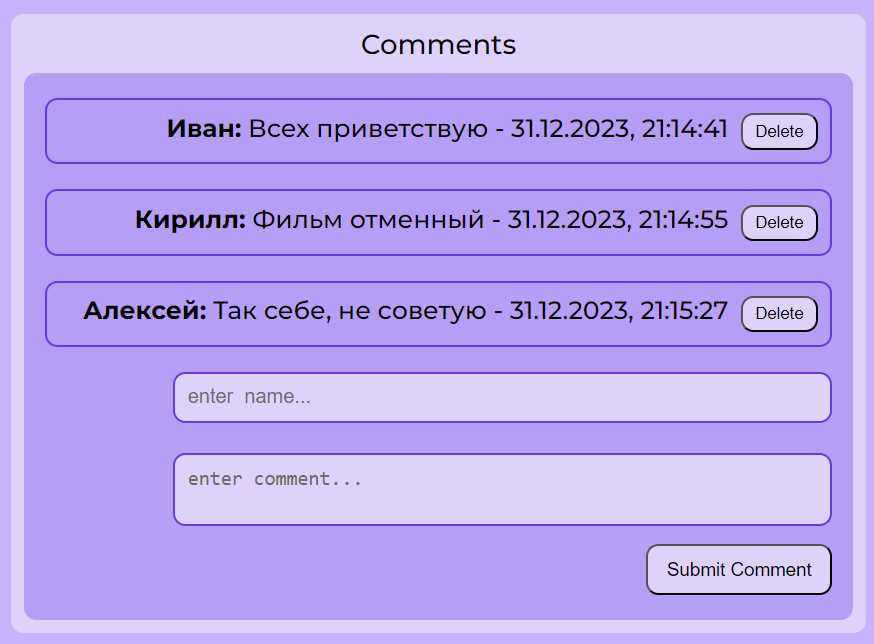


Рисунок 8 - отображение комментариев (светлая тема)

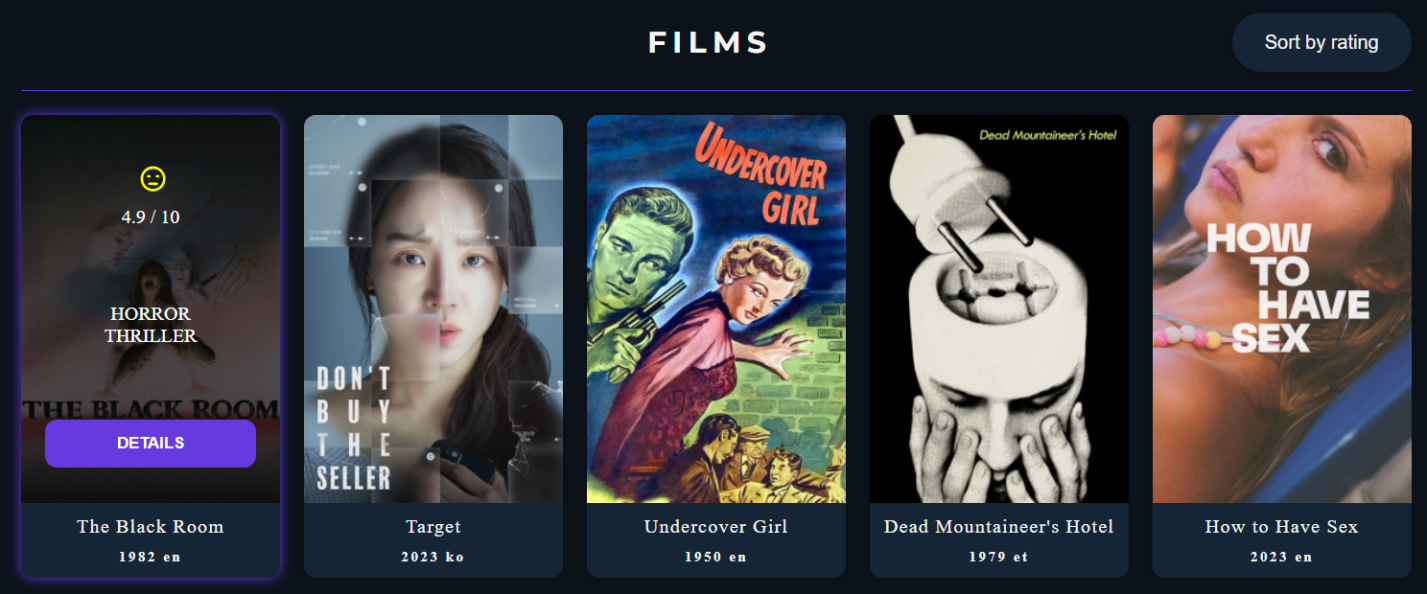


Рисунок 9 - список фильмов без сортировки по рейтингу

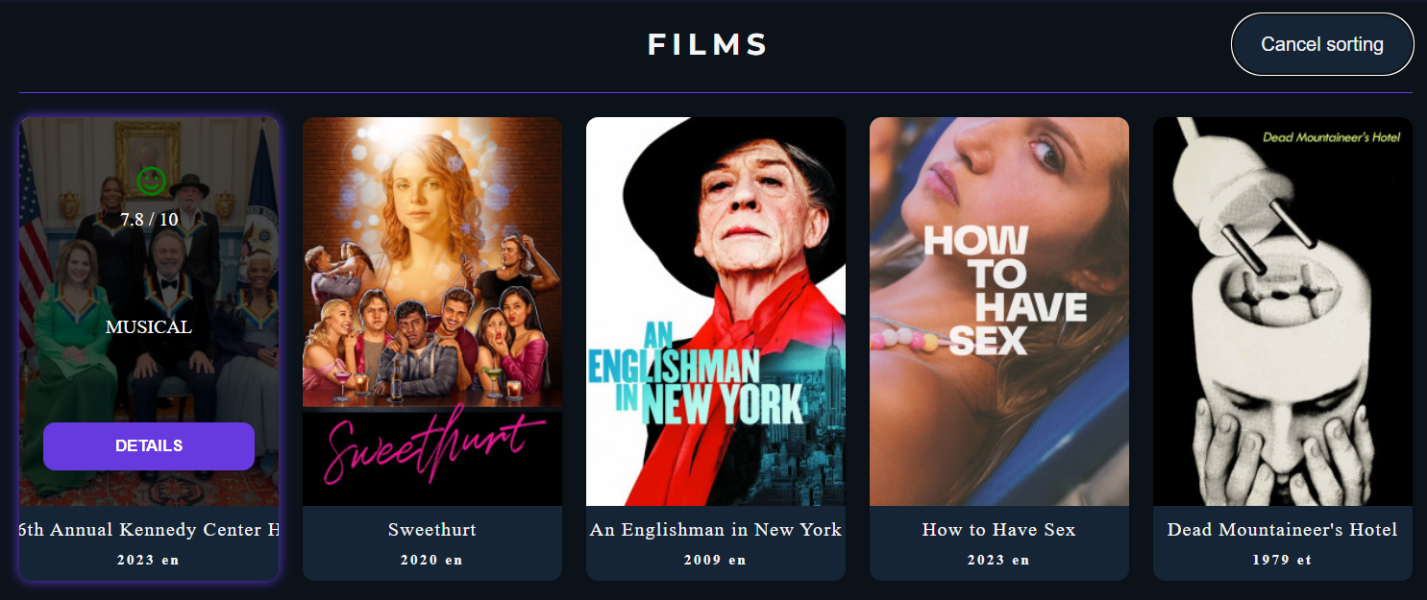


Рисунок 10 - список фильмов с сортировкой по рейтингу

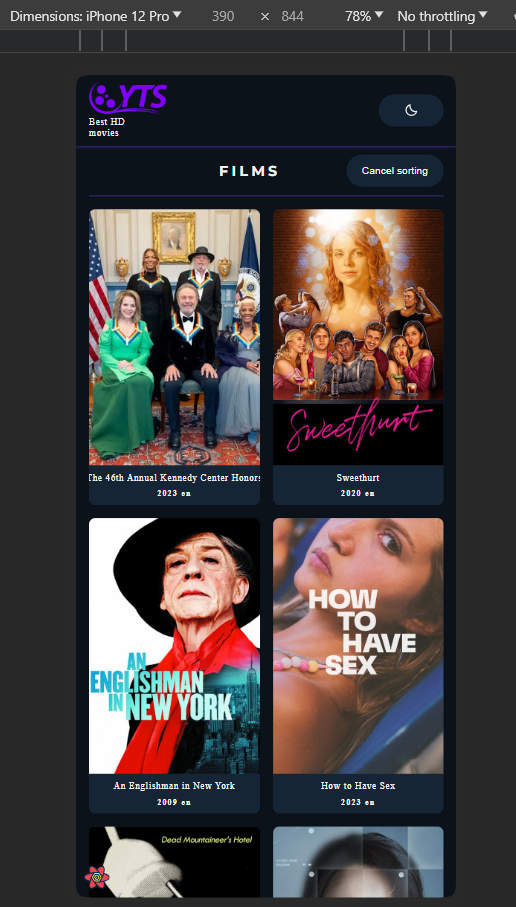


Рисунок 11 - адаптация главной страницы для телефонов (ч.1)

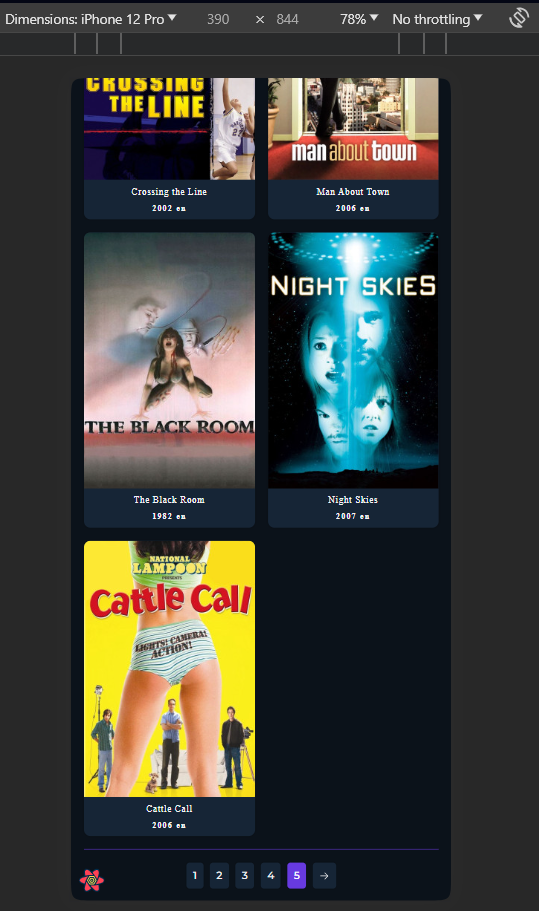


Рисунок 12 - адаптация главной страницы для телефонов (ч.2)

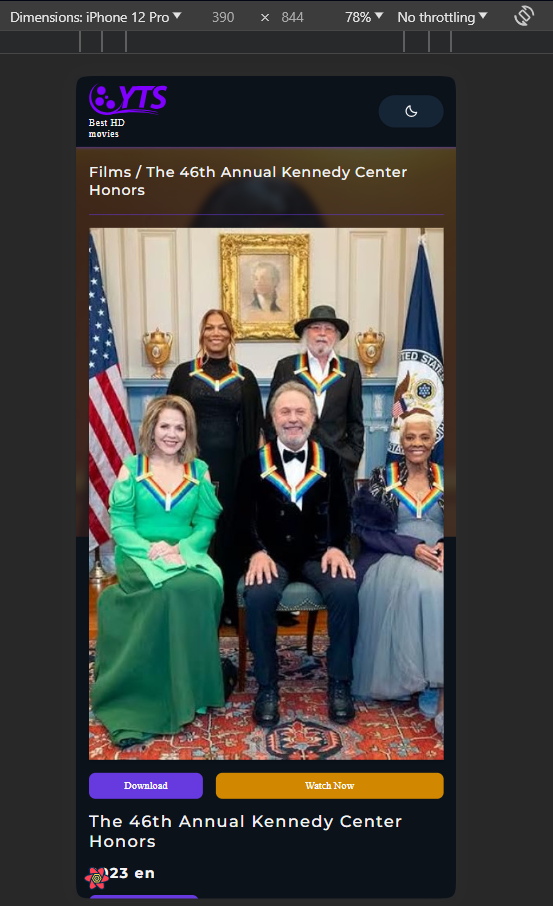


Рисунок 13 - адаптация страницы фильма для телефонов (ч.1)

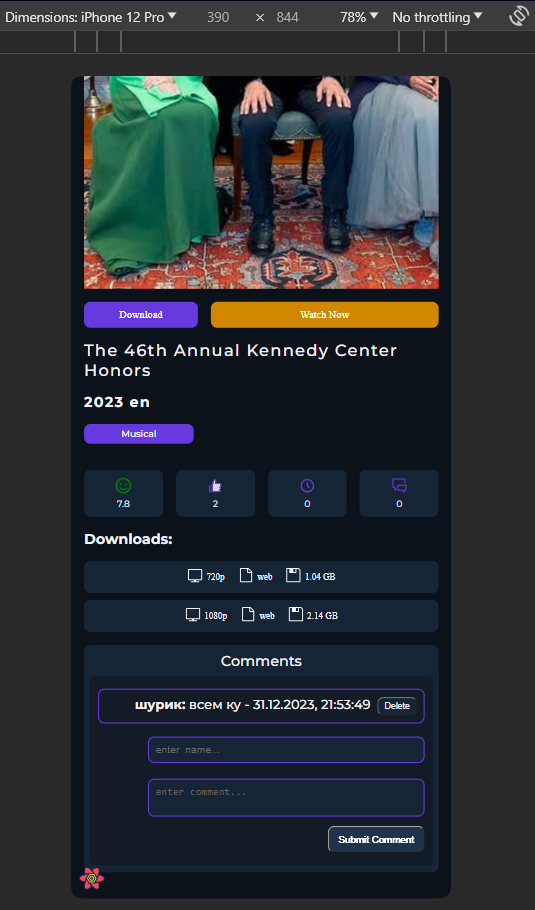


Рисунок 14 - адаптация страницы фильма для телефонов (ч.2)

# Структура проекта

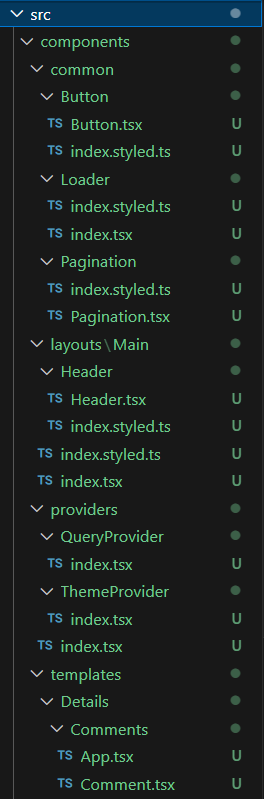


Рисунок 15 - структура проекта (ч.1)

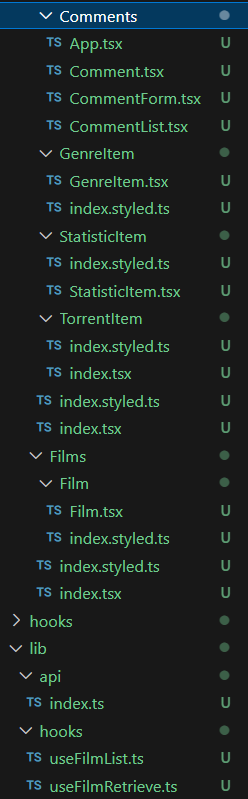


Рисунок 16- структура проекта (ч.2)

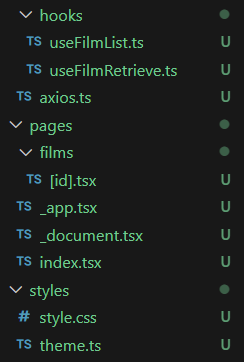


Рисунок 17- структура проекта (ч.3)

# Листинги кода

## Раздел комментариев

Листинг 1 - Comments.tsx

import React from 'react';

import styled from 'styled-components';

interface CommentProps {

  id: number;

  author: string;

  text: string;

  timestamp: string;

  onDelete: (id: number) => void;

}

const commentStyle: React.CSSProperties = {

  border: '2px solid #673ae0',

  padding: '10px',

  margin: '20px 0px',

  marginTop: '0px',

  borderRadius: '10px',

};

const textStyles: React.CSSProperties = {

  fontSize: '20px',

};

const StyledButton = styled.button`

  margin-left: 10px;

  color: ${props => props.theme.textColor};

  background: ${props => props.theme.iconColor};

  border-radius: 10px;

  padding: 5px 10px;

  cursor: pointer;

  box-shadow: 0;

  transition: filter 0.3s;

  &:hover {

    filter: brightness(1.4);

  }

`;

const Comment: React.FC<CommentProps> = ({ id, author, text, timestamp, onDelete }) => {

  return (

    <div style={commentStyle}>

      <strong>{author}:</strong> <span style={textStyles}>{text}</span> - {timestamp}

      <StyledButton onClick={() => onDelete(id)}>

        Delete

      </StyledButton>

    </div>

  );

};

export default Comment;

Листинг 2 - CommentForm.tsx

import React, { useState } from 'react';

import styled, { useTheme } from 'styled-components';

interface CommentFormProps {

  onCommentSubmit: (author: string, text: string, timestamp: string) => void;

}

const Input = styled.input<{ isFocused: boolean }>`

  color: ${(props) => props.theme.textColor};

  padding: 10px;

  font-size: 15px;

  width: 80%;

  border-radius: 10px;

  border: 2px solid #673ae0;

  background-color: ${(props) => props.theme.iconColor};

  transition: filter 0.3s;

  &:hover {

    filter: brightness(1.4);

  }

  &::placeholder {

    color: ${(props) => (props.isFocused ? 'transparent' : '#6b6b6b')};

    font-size: ${(props) => (props.isFocused ? '16px' : '16px')};

    // filter: ${(props) => (props.isFocused ? 'brightness(0.5)' : 'brightness(0.5)')};

  }

`;

const TextArea = styled.textarea<{ isFocused: boolean }>`

  color: ${(props) => props.theme.textColor};

  padding: 10px;

  font-size: 15px;

  width: 80%;

  resize: none;

  border-radius: 10px;

  border: 2px solid #673ae0;

  background-color: ${(props) => props.theme.iconColor};

  transition: filter 0.3s;

  &:hover {

    filter: brightness(1.4);

  }

  &::placeholder {

    color: ${(props) => (props.isFocused ? 'transparent' : '#6b6b6b')};

    font-size: ${(props) => (props.isFocused ? '16px' : '16px')};

    // filter: ${(props) => (props.isFocused ? 'brightness(0.5)' : 'brightness(0.5)')};

  }

`;

const SubmitButton = styled.button`

  // border: "none;

  border-radius: 10px;

  padding: 10px 15px;

  font-size: 15px;

  cursor: pointer;

  color: ${(props) => props.theme.textColor};

  background-color: ${(props) => props.theme.iconColor};

  margin-top: 10px;

  transition: filter 0.3s;

  &:hover {

    filter: brightness(1.4);

  }

`;

const CommentForm: React.FC<CommentFormProps> = ({ onCommentSubmit }) => {

  const [author, setAuthor] = useState('');

  const [text, setText] = useState('');

  const [isAuthorFocused, setAuthorFocused] = useState(false);

  const [isTextFocused, setTextFocused] = useState(false);

  const theme = useTheme();

  const handleAuthorFocus = () => {

    setAuthorFocused(true);

  };

  const handleAuthorBlur = () => {

    setAuthorFocused(false);

  };

  const handleTextFocus = () => {

    setTextFocused(true);

  };

  const handleTextBlur = () => {

    setTextFocused(false);

  };

  const handleSubmit = (e: React.FormEvent) => {

    e.preventDefault();

    if (author && text) {

      const timestamp = new Date().toLocaleString();

      onCommentSubmit(author, text, timestamp);

      setAuthor('');

      setText('');

    }

  };

  return (

    <form onSubmit={handleSubmit}>

      <label>

        <Input

          type="text"

          value={author}

          onChange={(e) => setAuthor(e.target.value)}

          onFocus={handleAuthorFocus}

          onBlur={handleAuthorBlur}

          required

          placeholder="enter  name..."

          isFocused={isAuthorFocused}

        />

        <br />

      </label>

      <br />

      <label>

        <TextArea

          value={text}

          onChange={(e) => setText(e.target.value)}

          onFocus={handleTextFocus}

          onBlur={handleTextBlur}

          required

          placeholder="enter comment..."

          isFocused={isTextFocused}

        />

      </label>

      <br />

      <SubmitButton type="submit">Submit Comment</SubmitButton>

    </form>

  );

};

export default CommentForm;

Листинг 3 - CommentList.tsx

import React from 'react';

import Comment from './Comment';

interface CommentData {

  id: number;

  author: string;

  text: string;

  timestamp: string;

}

interface CommentListProps {

  comments: CommentData[];

  onCommentDelete: (id: number) => void;

}

const CommentList: React.FC<CommentListProps> = ({ comments, onCommentDelete }) => {

  return (

    <div>

      {comments.length === 0 ? (

        <p style={{ textAlign: 'center' }}>No comments yet. Write the first comment!</p>

      ) : (

        comments.map((comment) => (

          <Comment

            key={comment.id}

            id={comment.id}

            author={comment.author}

            text={comment.text}

            timestamp={comment.timestamp}

            onDelete={onCommentDelete}

          />

        ))

      )}

    </div>

  );

};

export default CommentList;

Листинг 4 - App.tsx

import React, { useState } from 'react';

import CommentList from './CommentList';

import CommentForm from './CommentForm';

interface CommentData {

  id: number;

  author: string;

  text: string;

  timestamp: string;

}

const App: React.FC = () => {

  const [comments, setComments] = useState<CommentData[]>([]);

  const handleCommentSubmit = (author: string, text: string, timestamp: string) => {

    const newComment: CommentData = {

      id: Date.now(),

      author,

      text,

      timestamp,

    };

    setComments([...comments, newComment]);

    console.log(comments);

  };

  const handleCommentDelete = (id: number) => {

    const updatedComments = comments.filter((comment) => comment.id !== id);

    setComments(updatedComments);

  };

  return (

    <div>

      <CommentList comments={comments} onCommentDelete={handleCommentDelete} />

      <CommentForm onCommentSubmit={handleCommentSubmit} />

    </div>

  );

};

;

export default App;

## Карточки фильмов на главной странице

Листинг 5 - Film.tsx

import { MovieList } from "@/lib/api";

import Link from "next/link";

import React from "react";

import \* as Style from "./index.styled";

import { FaRegSmile } from "react-icons/fa";

import { CgSmileNone } from 'react-icons/cg';

import { CgSmileSad } from 'react-icons/cg';

const Film: React.FC<MovieList> = (props: MovieList) => {

  return (

    <Style.Film>

      <Link href={`/films/${props.id}`}>

        <Style.Content>

          <Style.Cards>

            <Style.Card1

              src={

                props.medium\_cover\_image

                  ? props.medium\_cover\_image

                  : "https://klike.net/uploads/posts/2019-05/1556708032\_1.jpg"

              }

              alt={props.medium\_cover\_image}

            ></Style.Card1>

            <Style.Card2>

              <Style.Rating>

                <div>

                  {(() => {

                    if (props.rating > 7) {

                      const coloricon = 'green';

                      return (

                        <Style.RatingIcon>

                          <FaRegSmile style={{ color: coloricon }} />

                        </Style.RatingIcon>

                      );

                    } else if (props.rating > 4.5) {

                      const coloricon = 'yellow';

                      return (

                        <Style.RatingIcon>

                          <CgSmileNone style={{ color: coloricon }} />

                        </Style.RatingIcon>

                      );

                    } else {

                      const coloricon = 'red';

                      return (

                        <Style.RatingIcon>

                          <CgSmileSad style={{ color: coloricon }} />

                        </Style.RatingIcon>

                      );

                    }

                  })()}

                </div>

                <Style.RatingText> {props.rating} / 10</Style.RatingText>

              </Style.Rating>

              {props.genres ? (

                <Style.Genres>

                  {0 in props.genres ? (

                    <Style.GenresText>

                      {props.genres[0].toUpperCase()}

                    </Style.GenresText>

                  ) : (

                    ""

                  )}

                  {1 in props.genres ? (

                    <Style.GenresText>

                      {props.genres[1].toUpperCase()}

                    </Style.GenresText>

                  ) : (

                    ""

                  )}

                  {2 in props.genres ? (

                    <Style.GenresText>

                      {props.genres[2].toUpperCase()}

                    </Style.GenresText>

                  ) : (

                    ""

                  )}

                </Style.Genres>

              ) : (

                ""

              )}

              <Style.Details>DETAILS</Style.Details>

            </Style.Card2>

          </Style.Cards>

          <Style.Name>{props.title}</Style.Name>

          <Style.Text>

            {props.year} {props.language}

          </Style.Text>

        </Style.Content>

      </Link>

    </Style.Film>

  );

};

export default Film;

Листинг 6 - index.styled.ts

// все про карточки в списке фильмов

import styled from "styled-components"

export const Card2 = styled.div`

    position: absolute;

    aspect-ratio: 1/1/5;

    display: flex;

    flex-direction: column;

    flex: 0 0 auto;

    width: 100%;

    height: 100%;

    object-fit: cover;

    box-sizing: border-box;

    background-color: rgba(0, 0, 0, 0.7);

    border-top-left-radius: 10px;

    border-top-right-radius: 10px;

    opacity: 0;

    pointer-events: none;

    transition: 0.3s;

    padding: 20px;

    color: white;

`;

export const Rating = styled.div`

    display: flex;

    flex: 0 0 auto;

    width: 100%;

    height: 30%;

    align-items: center;

    justify-content: center;

    flex-direction: column;

`;

export const RatingIcon = styled.div`

    display: flex;

    flex: 0 0 auto;

    width: 65%;

    align-items: center;

    justify-content: center;

    margin: 10px;

    color: ${props=> props.theme.ratingColor};

    font-size: ${props=> props.theme.textSizeTitle};

`;

export const RatingText = styled.div`

    display: flex;

    flex: 0 0 auto;

    width: 100%;

    align-items: center;

    justify-content: center;

    font-size: ${props=> props.theme.textSizeL};

`;

export const Genres = styled.div`

    width: 100%;

    height: 50%;

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

`;

export const GenresText = styled.div`

    width: 100%;

    display: flex;

    flex-direction: column;

    justify-content: center;

    align-items: center;

    font-weight: 500;

    font-size: ${props=> props.theme.textSizeL};

`;

export const Details = styled.button`

    cursor: pointer;

    border: 0;

    width: 100%;

    margin: 10px 0px;

    height: 40px;

    flex: 0 0 auto;

    border-radius: 10px;

    background-color:  ${props => props.theme.accentColor1};

    color: white;

    font-size: ${props=> props.theme.textSizeL};

    font-weight: 600;

    &:hover{

        background-color: rgb(138, 102, 237);

    }

`;

export const Film = styled.div`

    cursor: pointer;

    display: flex;

    flex-direction:column ;

    background-color: ${props => props.theme.iconColor};

    width: 100%;

    box-shadow: 0px 0px 5px ${props => props.theme.shadowColor};

    border-radius:10px;

    overflow: hidden;

    transition: all 0.3s ease;

    &:hover{

        box-shadow: 0px 0px 10px ${props => props.theme.accentColor1};

        transition: all 0.3s ease;

        & ${Card2}{

            transition: 0.3s;

            opacity: 1;

            pointer-events: all;

        }

    }

`;

export const Content = styled.div`

    display: flex;

    flex-direction: column;

    width: 100%;

    height: 100%;

    overflow: hidden;

`;

export const Cards = styled.div`

    position: relative;

    display: flex;

    flex-direction: column;

    width: 100%;

    overflow: hidden;

`;

export const Card1 = styled.img`

    aspect-ratio: 1/1/5;

    display: flex;

    flex: 0 0 auto;

    width: 100%;

    object-fit: cover;

    box-sizing: border-box;

    border-top-left-radius: 10px;

    border-top-right-radius: 10px;

`;

export const Name = styled.div`

    flex: 0 0 auto;

    display: flex;

    flex-wrap: nowrap;

    width: 100%;

    font-size: ${props => props.theme.textSizeTextM};

    letter-spacing: 1px;

    overflow: hidden;

    margin: 10px 0px 0px 0px;

    padding: 0px 10px;

    justify-content: center;

    box-sizing: border-box;

    text-overflow: ellipsis;

    white-space: nowrap;

    @media (max-width: 992px) and (min-width: 768px)  {

        font-size: ${props => props.theme.textSizeTextM900};

    }

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM500};

    }

`;

export const Text = styled.div`

    margin: 10px 0px 10px 0px;

    display:flex;

    flex: 0 0 auto;

    overflow: hidden;

    word-wrap: break-word;

    word-break: break-all;

    width: 100%;

    justify-content: center;

    font-weight: 600;

    letter-spacing: 2px;

    font-size: ${props => props.theme.textSizeTextS};

`;

## Рейтинг на странице сайта

Листинг 7 - index.styled.ts

import styled from 'styled-components';

export const StatisticItem = styled.div`

    aspect-ratio: 5/3;

    width: 100%;

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    background-color: ${props => props.theme.iconColor};

    box-shadow: 0px 0px 5px ${props => props.theme.shadowColor};

    border-radius: 10px;

`;

export const Icon = styled.div`

    color: ${props => props.theme.accentColor1};

    font-size: ${props => props.theme.textSizeTitle};

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle};

    }

`;

export const Text = styled.span`

    font-size: ${props => props.theme.textSizeTextM};

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM500};

    }

`;

Листинг 8 - StatisticItem.tsx

import React from "react";

import \* as Style from "./index.styled";

type StatisticItemProps = {

  icon: JSX.Element;

  text: number | string | undefined;

};

const StatisticItem: React.FC<StatisticItemProps> = (props) => {

  return (

    <Style.StatisticItem>

      <Style.Icon>{props.icon}</Style.Icon>

      <Style.Text>{props.text}</Style.Text>

    </Style.StatisticItem>

  );

};

export default StatisticItem;

Хэдер  
  
Листинг 9 - Header.tsx

import Link from "next/link";

import React from "react";

import { IoSunny } from "react-icons/io5";

import { BiMoon } from "react-icons/bi/";

import Button from "@/components/common/Button/Button";

import \* as Style from "./index.styled";

type HeaderProps = {

  switchTheme: () => void;

  theme: string;

};

const Header: React.FC<HeaderProps> = (props) => {

  return (

    <Style.Header id="header">

      <Style.Content>

        <Style.Logo>

        <Link href="/" passHref>

          <Style.Img src="https://svgur.com/i/11F0.svg"></Style.Img> </Link>

          <Style.Name>Best HD movies</Style.Name>

        </Style.Logo>

          <Style.SearchArea>

          </Style.SearchArea>

          <Style.ButtonSwitchTheme>

          <Button

            name={props.theme === "dark" ? <BiMoon /> : <IoSunny />}

            action={props.switchTheme}

          />

        </Style.ButtonSwitchTheme>

      </Style.Content>

    </Style.Header>

  );

};

export default Header;

Листинг 10 - index.styled.tsx

import styled from 'styled-components';

export const Header = styled.header`

    display: flex;

    width: 100%;

    // min-width: 200px;

    min-height: 80px;

    // align-items: center;

    justify-content: center;

    color: ${props => props.theme.textColor};

    box-shadow: 0px 0px 6px ${props => props.theme.accentColor1};

    background-color: ${props => props.theme.backgroundColor};

    position: sticky;

    overflow: hidden;

    z-index: 50;

    @media (max-width: 768px) and (min-width: 576px)  {

        // min-height: 60px;

    }

    @media (max-width: 576px) and (min-width: 324px) {

        width: 100%;

    }

`;

export const Content = styled.div`

    display: flex;

    align-items: center;

    width: 100%;

    height: 100%;

    padding: 0px 20px;

    background-color: ${props => props.theme.backgroundColor};

    box-sizing: border-box;

    @media (min-width: 1200px){

        width: 1200px;

    }

    @media (max-width: 1200px) and (min-width: 992px)  {

        width: 100%;

    }

    @media (max-width: 992px) and (min-width: 768px)  {

        width: 100%;

    }

    @media (max-width: 768px) and (min-width: 576px)  {

        width: 100%;

    }

    @media (max-width: 576px) {

        width: max-width;

        }

`;

export const Logo = styled.div`

    display: flex;

    width: 12%;

    flex-direction: column;

    align-items: flex-start;

    margin: 10px 0px;

    &:hover {

        filter: brightness(1.3);

        cursor: pointer;

    }

`;

export const Img = styled.img`

    display: flex;

    height: 50px;

`;

export const Name = styled.span`

    display: flex;

    align-items: center;

    font-size: ${props => props.theme.textSizeTextM};

    font-weight: 500;

    font-family: Roboto;

    letter-spacing: 1px;

    margin: 5px 0px 0px 0px;

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM500};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM500};

    }

`;

export const ButtonSwitchTheme = styled.div`

    display: flex;

`;

export const SearchArea = styled.div`

    display: flex;

    width: 80%;

    height: 100%;

    padding: 0px 20px;

    align-items: right;

`;

export const SearchInput = styled.input`

    display: flex;

    width: 60%;

    padding: 8px;

    margin: 3px 0px;

    background-color: ${props => props.theme.iconColor};

    border: 1px solid ${props => props.theme.accentColor1};

    border-radius: 10px;

    color: ${props => props.theme.textColor};

    font-size: 14px;

`;

export const SearchButton = styled.button`

    display: flex;

    padding: 8px;

    margin-left: 15px;

    background-color: ${props => props.theme.accentColor1};

    color: ${props => props.theme.textColor};

    border: none;

    align-items: center;

    justify-content: center;

    font-size: 18px;

    box-shadow: 0px 0px 5px ${props => props.theme.shadowColor} ;

    border-radius: 50px;

    cursor: pointer;

    width: 100px;

    height: 50px;

    &:hover {

        filter: brightness(1.3);

    }

`;

## Статистика на странице фильма

Листинг 11 - StatisticItem

import React from "react";

import \* as Style from "./index.styled";

type StatisticItemProps = {

  icon: JSX.Element;

  text: number | string | undefined;

};

const StatisticItem: React.FC<StatisticItemProps> = (props) => {

  return (

    <Style.StatisticItem>

      <Style.Icon>{props.icon}</Style.Icon>

      <Style.Text>{props.text}</Style.Text>

    </Style.StatisticItem>

  );

};

export default StatisticItem;

Листинг 12 - index.styled.ts

import styled from 'styled-components';

export const StatisticItem = styled.div`

    aspect-ratio: 5/3;

    width: 100%;

    display: flex;

    flex-direction: column;

    align-items: center;

    justify-content: center;

    background-color: ${props => props.theme.iconColor};

    box-shadow: 0px 0px 5px ${props => props.theme.shadowColor};

    border-radius: 10px;

`;

export const Icon = styled.div`

    color: ${props => props.theme.accentColor1};

    font-size: ${props => props.theme.textSizeTitle};

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle};

    }

`

export const Text = styled.span`

    font-size: ${props => props.theme.textSizeTextM};

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTextM500};

    }

`;

## Страница фильма

Листинг 13 - index.tsx

import Loader from "@/components/common/Loader";

import \* as Style from "@/components/templates/Details/index.styled";

import { useFilmRetrieve } from "@/lib/hooks/useFilmRetrieve";

import { useRouter } from "next/router";

import { VscCommentDiscussion} from "react-icons/vsc";

import GenreItem from "./GenreItem/GenreItem";

import StatisticItem from "./StatisticItem/StatisticItem";

import Torrent from "./TorrentItem";

import { AiTwotoneLike } from "react-icons/ai";

import { FaRegSmile } from "react-icons/fa";

import { CgSmileNone, CgSmileSad } from 'react-icons/cg';

import { BiTimeFive } from "react-icons/bi";

import App from "../../templates/Details/Comments/App";

import { useState } from "react";

const Details = () => {

  const router = useRouter();

  const { filmRetrieve, isLoading } = useFilmRetrieve(

    (router.query.id as string) || ""

  );

  const [commentsCount, setCommentsCount] = useState(0);

  if (isLoading || !filmRetrieve?.data?.movie) {

    return <Loader />;

  }

  const genresList = filmRetrieve?.data.movie.genres.map((value) => {

    return <GenreItem key={value} text={value} />;

  });

  const torrentsList = filmRetrieve?.data.movie.torrents.map((item, index) => {

    return (

      <Torrent

        key={index}

        href={item.url}

        quality={item.quality}

        type={item.type}

        size={item.size}

      />

    );

  });

  const filmRate: number = filmRetrieve?.data.movie.rating as number;

  const getIconStyle = (filmRate: number) => {

    if (filmRate > 7) {

      return { color: 'green' };

    } else if (filmRate > 4.5) {

      return { color: 'yellow' };

    } else {

      return { color: 'red' };

    }

  };

  return (

    <Style.Details>

      <Style.BackgroundImage

        src={filmRetrieve?.data.movie.background\_image\_original}

      ></Style.BackgroundImage>

      <Style.Content>

        <Style.ContentTitle>

        Films / {filmRetrieve?.data.movie.title}

        </Style.ContentTitle>

        <Style.Data>

          <Style.Image>

            <Style.Img

              src={filmRetrieve?.data.movie.large\_cover\_image}

            ></Style.Img>

            <Style.Buttons>

              <Style.DownloadButton href={filmRetrieve?.data.movie.url}>

                Download

              </Style.DownloadButton>

              <Style.WatchButton href={filmRetrieve?.data.movie.url}>

                Watch Now

              </Style.WatchButton>

            </Style.Buttons>

          </Style.Image>

          <Style.Description>

            <Style.Title>{filmRetrieve?.data.movie.title}</Style.Title>

            <Style.Year>

              {filmRetrieve?.data.movie.year +

                " " +

                filmRetrieve?.data.movie.language}

            </Style.Year>

            <Style.Genres>{genresList}</Style.Genres>

            <Style.DescriptionFull>

              {filmRetrieve?.data.movie.description\_full}

            </Style.DescriptionFull>

            <Style.Statistic>

              <div>

                {(() => {

                  if (filmRate > 7) {

                    return (

                      <StatisticItem

                        icon={<FaRegSmile style={getIconStyle(filmRate)} />}

                        text={filmRetrieve?.data.movie.rating}

                      ></StatisticItem>

                    );

                  } else if (filmRate > 4.5) {

                    return (

                      <StatisticItem

                        icon={<CgSmileNone style={getIconStyle(filmRate)} />}

                        text={filmRetrieve?.data.movie.rating}

                      ></StatisticItem>

                    );

                  } else {

                    return (

                      <StatisticItem

                        icon={<CgSmileSad style={getIconStyle(filmRate)} />}

                        text={filmRetrieve?.data.movie.rating}

                      ></StatisticItem>

                    );

                  }

                })()}

              </div>

              <StatisticItem

                icon={<AiTwotoneLike />}

                text={filmRetrieve?.data.movie.like\_count}

              ></StatisticItem>

              <StatisticItem

                icon={<BiTimeFive />}

                text={filmRetrieve?.data.movie.runtime}

              ></StatisticItem>

              <StatisticItem

                icon={<VscCommentDiscussion />}

                text={commentsCount}

              ></StatisticItem>

            </Style.Statistic>

            <Style.TorrentsTitle>Downloads:</Style.TorrentsTitle>

            <Style.Torrents>{torrentsList}</Style.Torrents>

            <Style.CommentsAll>

              Comments

              <Style.CommentsWorkspace>

                <Style.CommentsWorkspace1>

                  <App />

                </Style.CommentsWorkspace1>

              </Style.CommentsWorkspace>

            </Style.CommentsAll>

          </Style.Description>

        </Style.Data>

      </Style.Content>

    </Style.Details>

  );

};

export default Details;

Листинг 14 - index.styled.ts

import styled from 'styled-components';

export const Details = styled.section`

    display: flex;

    width: 100%;

    height: 100%;

    flex: 0 0 auto;

    justify-content: center;

    background-color: ${props => props.theme.backgroundColor};

    transition: 0.1s;

    position: relative;

    @media (max-width: 768px) and (min-width: 576px)  {

        width: 100%;

    }

    @media (max-width: 576px) and (min-width: 324px) {

        width: 100%;

        grid-template-columns: 1fr;

    }

`;

export const Content = styled.div`

    display: flex;

    flex-direction: column;

    width: 1000px;

    padding: 0px 20px;

    margin: 0px 0px;

    box-sizing: border-box;

    z-index: 1;

    @media (min-width: 1200px){

        width: 1200px;

    }

    @media (max-width: 1200px) and (min-width: 992px)  {

        width: 100%;

    }

    @media (max-width: 992px) and (min-width: 768px)  {

        padding: 0px 20px;

        margin: 0px 0px;

        width: 100%;

    }

    @media (max-width: 768px) and (min-width: 576px)  {

        // padding: 0px 20px;

        margin: 0px 0px;

        width: 100%;

    }

    @media (max-width: 576px) {

        width: 100%;

    }

`;

export const ContentTitle = styled.span`

    letter-spacing: 1px;

    font-weight: 500;

    display: flex;

    flex: 0 0 auto;

    width: 100%;

    margin: 25px 0px;

    align-items: center;

    box-sizing: border-box;

    font-size: ${props => props.theme.textSizeTitle};

    @media (max-width: 768px) and (min-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle700};

    }

    @media (max-width: 576px)  {

        font-size: ${props => props.theme.textSizeTitle500};

        width:100%

    }

    & a{

        font-family: Roboto;

        font-size:  ${props => props.theme.textSizeTitle};

        font-weight: 500;

        letter-spacing: 1px;

        transition: 0.3s;

        &:hover{

            transition: 0.3s;

            color: ${props => props.theme.accentColor1};

        }

    }

`;

export const Data = styled.div`

    display: grid;

    width: 100%;

    // align-items: center;

    // align-self: center;

    grid-template-columns: 2fr 3fr;

    grid-gap: 20px;

    border-top: 1px solid ${props => props.theme.accentColor1};

    border-bottom: 1px solid ${props => props.theme.accentColor1};

    padding: 20px 0px;

    box-sizing: border-box;

    @media (min-width: 1200px){

        width: 100%;

    }

    @media (max-width: 1200px) and (min-width: 992px)  {

        width: 100%;

    }

    @media (max-width: 992px) and (min-width: 768px)  {

        width: 100%;

    }

    @media (max-width: 768px) and (min-width: 576px)  {

        width: 100%;

        grid-template-columns: 1fr;

    }

    @media (max-width: 576px)  {

        width: 100%;

        grid-template-columns: 2fr;

    }

`;

export const BackgroundImage = styled.img`

    position: absolute;

    top: 0;

    width: 100%;

    height: 600px;

    object-fit: cover;

    opacity: 0.3;

    @media (max-width: 768px) and (min-width: 576px)  {

        height: 500px;

        width: 100%;

    }

    @media (max-width: 576px) {

            width: 100%;

    }

`;

export const Image = styled.div`

    display: flex;

    width: 100%;

    height: 100%;

    flex-direction: column;

`;

export const Img = styled.img`

    aspect-ratio: 2/3

    display: grid;

    width: 100%;

    object-fit: cover;

    box-sizing: border-box;

    transition: 0.5s;

    box-shadow: 0px 3px 5px ${props=>props.theme.shadowColor};;

    margin: 0px 0px 20px 0px;

    @media (max-width: 768px)  {

    }

`;

export const Buttons = styled.div`

    display: grid;

    grid-template-columns: 1fr 2fr;

    grid-gap: 20px;

    @media (max-width: 768px) and (min-width: 576px)  {

        grid-template-columns: 1fr 2fr;

    }

    @media (max-width: 576px)  {

        grid-template-columns: 1fr 2fr;

    }

`;

export const Button = styled.a`

    cursor: pointer;

    display: flex;

    width: 100%;

    height: 40px;

    align-items: center;

    justify-content: center;

    color:white;

    background-color: white;

    border: 0px;

    border-radius:10px;

    font-size: ${props=>props.theme.textSizeTextM};

    &:hover{

        opacity: 0.7;

    }

`;

export const DownloadButton = styled(Button)`

    background-color: ${props=>props.theme.accentColor1};

`;

export const WatchButton = styled(Button)`

    background-color:${props=>props.theme.accentColor2};

`;

export const Description = styled.div`

    display: grid;

    align-items: flex-start;

    align-self: flex-start;

    grid-template-columns: 1fr;

    grid-gap: 20px;

`;

export const Title = styled.div`

    display: flex;

    font-size: ${props=> props.theme.textSizeTitle};

    letter-spacing: 2px;

    font-weight: 500;

`;

export const Year = styled.div`

    display: flex;

    width: 100%;

    flex: 0 0 auto;

    font-weight: 600;

    letter-spacing: 2px;

    font-size: ${props=> props.theme.textSizeTextL};

    color: ${props=> props.theme.textColor};

`;

export const DescriptionFull = styled.div`

    display: flex;

    font-size: ${props=> props.theme.textSizeTextM};

    line-height: 25px;

    letter-spacing: 1px;

    text-align: justify;

`;

export const Genres = styled.div`

    display: grid;

    font-size: ${props=> props.theme.textSizeTextM};

    grid-template-columns: repeat(3,1fr);

    grid-template-columns: repeat(auto-fill, minmax(20%, 1fr));

    grid-gap: 20px;

    @media (max-width: 768px) and (min-width: 576px)  {

        grid-template-columns: 1fr 1fr 1fr;

    }

    @media (max-width: 576px)  {

        grid-template-columns: 1fr 1fr 1fr;

    }

`;

export const Statistic = styled.div`

    display: grid;

    grid-template-columns: repeat(4, 1fr);

    grid-gap: 35px;

    width: 100%;

    @media (max-width: 1200px) and (min-width: 768px)  {

        grid-gap: 25px;

    }

    @media (max-width: 768px) and (min-width: 576px)  {

        grid-gap: 20px;

    }

    @media (max-width: 576px)  {

        grid-gap: 20px;

    }

`;

export const TorrentsTitle = styled.div`

    display: flex;

    width: 100%;

    flex: 0 0 auto;

    font-weight: 600;

    font-size: ${props=> props.theme.textSizeTextL};

    color: ${props=> props.theme.textColor};

`;

export const Torrents = styled.div`

    display: grid;

    grid-template-columns: 1fr 1fr;

    grid-gap: 30px;

    aligh-text: center;

    font-size: 16px;

    @media (max-width: 1200px) and (min-width: 576px)  {

        grid-gap: 20px;

        grid-template-columns: 1fr;

    }

    @media (max-width: 576px)  {

        grid-template-columns: 1fr;

        grid-gap: 10px;

    }

`;

export const CommentsAll = styled.div`

display: inline-block;

    height: auto;

    margin: 0px 0px 30px 0px;

    width: 100%;

    text-align: center;

    margin-left:auto;

    font-weight: 500;

    font-size: ${props=> props.theme.textSizeTextL};

    background-color: ${props => props.theme.iconColor};

    border-radius: 10px;

    padding: 10px 0px;

`;

export const CommentsWorkspace = styled.div`

    display: inline-block;

    width: 97%;

    height: auto;

    margin-top: 10px;

    background-color: ${props=> props.theme.commentBackColor};

    border-radius: 10px;

    text-align: middle;

    font-weight: 500;

`;

export const CommentsWorkspace1 = styled.div`

    display: inline-block;

    border-radius: 10px;

    width: 95%;

    height: auto;

    margin-top: 20px;

    margin-bottom: 20px;

    text-align: right;

    height: auto;

    font-size: 20px;

`;

export const CommentsWorkspace2 = styled.div`

    display: inline-block;

    border-radius: 10px;

    background: green;

    width: 95%;

    margin-top: 10px;

    text-align: center;

    height: 40px;

`;

# Вывод

В ходе работы был реализован киносайт с использованием next.js и tailwind. Сайт получает список фильмов по API и визуализирует полученные данные. К каждому фильму из списка имеется возможность оставлять и удалять комментарии. Имеется адаптив сайта для устройств с меньшим размером экрана.  
 Ссылка на git-репозиторий: **<https://github.com/adzhisa/UI-UX>**