Московский государственный технический университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Отчёт по домашнему заданию
по курсу «Разработка интернет-приложений»

Выполнил:	Проверил:
студент группы ИУ5-51Б	преподаватель каф. ИУ5
Головацкий А. Д.	Гапанюк Ю. Е.
Подпись и дата:	Подпись и дата:
15.12.2021	

Москва, 2021 г.

Описание задания.

1. Разработать сервер на node js, используя express. В качестве базы данных использовать MongoDB. Веб-сервис развернуть на облачном сервисе heroku.
2. Разработать SPA, используя React + TypeScript. В качестве state-менеджера использовать Redux.
3. Реализовать тестирование, используя: скриншотные тесты, unit-тесты, интеграционные тесты (поднимать selenium и прогонять на поднятом браузере приложение).
4. Добавить авторизацию, используя FireBase.

Текст программы.

Сервер:

server/server.js:

```
const mongoose = require('mongoose');
require('./app/models/movie')

const express = require('express');
const config = require('./config')
const app = express();
config.express(app);
config.routes(app);

const {port, mongoUri} = config.app;

mongoose.connect(mongoUri)
    .then(() => app.listen(
        port,
        () => console.log(`Server is running on: http://localhost:${port}`)
    ))
    .catch(() => console.log(`Error connecting to mongodb: ${mongoUri}`));
```

server/app/controllers/movies.js:

```
const mongoose = require('mongoose');

const Movie = mongoose.model('Movie');

const getAll = (req, res) => {
    Movie.find({"title": new RegExp('.*' + req.params.search + '.*')})
        .exec()
        .then(movies => {
            if (movies.length)
                res.json(movies)
            else
                res.status(404).json(movies);
        })
        .catch(error => res.status(500).json(error));
}

const create = (req, res) => {
    Movie.create(req.body)
        .then(created => res.json(created))
        .catch(error => res.status(500).json(error));
}
```

```
const update = (req, res) => {
    Movie.findOneAndUpdate({id: req.params.id}, req.body)
        .exec()
        .then(movie => res.json(movie))
        .catch(error => res.status(500).json(error));
}

const remove = (req, res) => {
    Movie.deleteOne({"title": new RegExp('.*' + req.params.title + '.*')})
        .exec()
        .then(() => res.json({success: true}))
        .catch(error => res.status(500).json(error));
}

module.exports = {
    getAll,
    create,
    update,
    remove
}
```

server/app/models/movie.js:

```
const mongoose = require('mongoose');

const MovieSchema = new mongoose.Schema({
    id: Number,
    adult: Boolean,
    backdrop_path: String,
    genre_ids: Array,
    original_language: String,
    original_title: String,
    overview: String,
    popularity: Number,
    poster_path: String,
    title: String,
    video: Boolean,
    vote_average: Number,
    vote_count: Number,
})

const Movie = mongoose.model('Movie', MovieSchema);
```

server/config/app.js:

```
module.exports = {
    port: process.env.port || 3001,
    mongoUri: 'mongodb://localhost:27017/movie-db'
}
```

server/config/express.js:

```
const bodyParser = require('body-parser');

module.exports = (app) => {
    app.use(bodyParser.json());
    app.use(function (req, res, next) {
        res.setHeader('Access-Control-Allow-Origin', 'http://localhost:3000');
        res.setHeader('Access-Control-Allow-Methods', 'GET, POST, OPTIONS, PUT,
PATCH, DELETE');
        res.setHeader('Access-Control-Allow-Headers', 'X-Requested-With,content-
type');
        res.setHeader('Access-Control-Allow-Credentials', true);
        next();
    });
}
```

server/config/express.js:

```
const app = require('./app');
const express = require('./express');
const routes = require('./routes');

module.exports = {
    app,
    express,
    routes
}
```

server/config/routes.js:

```
const movies = require('../app/controlers/movies');

module.exports = (app) => {
    app.get('/', (req, res) => res.send('q world'));
    app.get('/movies/:search', movies.getAll);
    app.post('/movies/', movies.create);
    app.put('/movies/:id', movies.update);
    app.delete('/movies/:title', movies.remove);
}
```

server/Procfile (для деплоя на heroku):

```
web: node server.js
```

Клиент:

src/App.tsx:

```
import './App.css';
import Routes from './Routes';

function App() {
    return (
        <div className="App">
            <Routes />
        </div>
    );
}

export default App;
```

src/Routes.tsx:

```
import React from 'react';
import { BrowserRouter, Route, Switch } from 'react-router-dom';
import SearchPage from './components/SearchPage/SearchPage'
import MovieInfo from './components/MovieInfo/MovieInfo'
import MovieNotFound from './components/MovieNotFound/MovieNotFound'
import {Login} from "./components/Login/Login";
import {PrivateRoute} from "./components/PrivateRoute";

const Routes = () => (
    <BrowserRouter>
        <Switch>
            <Route exact path="/auth" component={Login} />
            <PrivateRoute exact path="/" component={SearchPage} />
            <PrivateRoute exact path="/movieInfo/:movieName"
component={MovieInfo} />
            <PrivateRoute exact path="/movieNotFound" component={MovieNotFound}
/>
        </Switch>
    </BrowserRouter>
);

export default Routes;
```

src/utils.ts:

```
export function normalizeUrl(url: string): string {
    if ((url.split(''))[url.length - 1] === '/')
        return (url.slice(0, -1));
    return (url);
}

export function handleTextHighlight(title: string, input:
React.RefObject<HTMLInputElement>) {
    let search_str: string = '';
    if (input.current?.value !== undefined)
        search_str = input.current?.value.trim();
    let ind: number = title.toLowerCase().indexOf(search_str.toLowerCase());
    if (ind === -1) {
        return ({
            before: title,
            search_str: '',
            after: ''
        });
    }
    let before: string = title.split('').slice(0, ind).join('');
    search_str = title.split('').slice(ind, ind + search_str.length).join('');
    let after: string = title.split('').slice(ind + search_str.length,
title.length).join('');
    return ({
        before: before,
        search_str: search_str,
        after: after,
    });
}
```

src/App.css:

```
.App {
    min-width: 320px;
    padding: 20px;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}
```

src/index.ts:

```
import React, {createContext} from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { store } from './store';
import { Provider } from 'react-redux';
// Import the functions you need from the SDKs you need
import firebase from "firebase/compat";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
    apiKey: "AIzaSyDWZxj7oKY7JikN2XiYKl3uxBsLG9fXa3I",
    authDomain: "movie-react-235fb.firebaseapp.com",
    projectId: "movie-react-235fb",
    storageBucket: "movie-react-235fb.appspot.com",
    messagingSenderId: "933630337933",
    appId: "1:933630337933:web:2097c19ef8c01b68c6d3c4",
    measurementId: "G-3J0XMJVVWL"
};

firebase.initializeApp(firebaseConfig);

const auth = firebase.auth();
const firestore = firebase.firestore();

export const Context = createContext<any>(null)

ReactDOM.render(
  <React.StrictMode>
      <Context.Provider value={{
          firebase,
          auth,
          firestore
      }}>
        <Provider store={store}>
          <App />
        </Provider>
      </Context.Provider>
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

src/actions/movie.ts:

```
import { Dispatch } from "redux"
import { MovieAction, MovieActionType } from "../types/types"
import axios from "axios";

export const fetchMovie = (search_str: string) => {
    return async (dispatch: Dispatch<MovieAction>) => {
        axios.get(`http://localhost:3001/movies/${search_str}`)
            .then((res) => {
                dispatch({
                    type: MovieActionType.FETCH_MOVIE_SUCCESS,
                    payload: res.data
                })
            })
            .catch(() => {
                dispatch({
                    type: MovieActionType.FETCH_MOVIE_ERROR,
                    payload: 404
                })
            })
    }
}

export const addMovie = (movie: any) => {
    return (dispatch: Dispatch<MovieAction>) => {
        dispatch({
            type: MovieActionType.ADD_MOVIE,
            payload: movie
        })
    }
}

export const clearMovie = () => {
    return (dispatch: Dispatch<MovieAction>) => {
        dispatch({
            type: MovieActionType.CLEAR_MOVIE,
            payload: []
        })
    }
}
```

src/actions/user.ts:

```
import {Dispatch} from "redux";
import {UserAction, UserActionType} from "../types/types";

export const addUser = (user: any) => {
    return (dispatch: Dispatch<UserAction>) => {
        dispatch({
            type: UserActionType.ADD_USER,
            payload: user
        })
    }
}
```

src/hooks/useMovieActions.ts:

```
import {useDispatch} from 'react-redux';
import {bindActionCreators} from 'redux';
import * as movieActionCreators from '../actions/movie';

export const useMovieActions = () => {
    const dispatch = useDispatch();
    return bindActionCreators(movieActionCreators, dispatch);
}
```

src/hooks/useTypedSelector.ts:

```
import {useSelector, TypedUseSelectorHook} from 'react-redux'
import { RootState } from '../store';

export const useTypedSelector: TypedUseSelectorHook<RootState> = useSelector;
```

src/hooks/useUserActions.ts:

```
import {useDispatch} from "react-redux";
import {bindActionCreators} from "redux";
import * as userActionCreators from "../actions/user";

export const useUserActions = () => {
    const dispatch = useDispatch();
    return bindActionCreators(userActionCreators, dispatch);
}
```

src/store/index.ts:

```
import { createStore, applyMiddleware } from 'redux';
import thunk from 'redux-thunk';
import {combineReducers} from 'redux';
import { movieReducer } from './reducers/movie';
import { userReducer } from "./reducers/user";

const rootReducer: any = combineReducers({
    movie: movieReducer,
    user: userReducer,
})

export const store = createStore(rootReducer, applyMiddleware(thunk));

export type RootState = ReturnType<typeof rootReducer>
```

src/store/reducers/movie.ts:

```
import {MovieAction, MovieActionType, MovieState} from '../../types/types'

const defaultState: MovieState = {
    movie: null,
}

const Reducer = (state: MovieState = defaultState, action: MovieAction) => {
    switch (action.type) {
        case MovieActionType.FETCH_MOVIE_SUCCESS: {
            return {movie: action.payload}
        }
        case MovieActionType.FETCH_MOVIE_ERROR: {
            return {movie: action.payload}
        }
        case MovieActionType.CLEAR_MOVIE: {
            return {movie: action.payload}
        }
        case MovieActionType.ADD_MOVIE: {
            return {movie: state.movie.concat(action.payload)};
        }
        default: {
            return state;
        }
    }
}

export {Reducer as movieReducer};
```

src/store/reducers/user.ts:

```
import { UserAction, UserActionType, UserState} from "../../types/types";

const defaultState: UserState = {
    user: null,
}

const reducer = (state: UserState = defaultState, action: UserAction) => {
    switch (action.type) {
        case UserActionType.ADD_USER: {
            return {user: action.payload}
        }
        default: {
            return state;
        }
    }
}

export {reducer as userReducer};
```

src/types/types.ts:

```
export interface MovieState {
    movie: any;
}

export interface UserState {
    user: any;
}


export enum MovieActionType {
    FETCH_MOVIE_SUCCESS,
    FETCH_MOVIE_ERROR,
    CLEAR_MOVIE,
    ADD_MOVIE
}

export enum UserActionType {
    ADD_USER,
}

export interface MovieAction {
    type: MovieActionType,
    payload: any,
}

export interface UserAction {
    type: UserActionType;
    payload: any;
}

export type ParamsType = { movieName: string };
```

src/components/Login/Login.tsx:

```
import React, {useContext} from 'react'
import './Login.css'
import {useHistory} from "react-router";
import {Context} from "../../index";
import firebase from "firebase/compat";
import {useUserActions} from "../../hooks/useUserActions";
import {Redirect} from "react-router-dom";

export const Login: React.FC = () => {
    const history = useHistory();
    const {auth} = useContext(Context);
    const {addUser} = useUserActions();

    const login = async () => {
        const provider = new firebase.auth.GoogleAuthProvider();
        const {user} = await auth.signInWithPopup(provider)
        addUser(user);
        localStorage.setItem('user', JSON.stringify(user?.uid));
    }

    const handleClick = () => {
        login()
            .then(() => history.push('/'))
            .catch(error => alert(error));
    }

    if (localStorage.getItem('user'))
        return (<Redirect exact to='/' />)

    return (
        <div className="wrapper">
            <div className="content">
```

```
                <button className="login-btn" onClick={handleClick}>
                    Login
                </button>
            </div>
        </div>
    )
}
```

src/components/Login/Login.css:

```
.wrapper {
  margin-top: 100px;

}

.login-btn {
  width: 200px;
  height: 100px;
  font-size: 32px;
  border-radius: 10px;
  border: 1px solid blueviolet;
}

.content {
  display: flex;
  align-items: center;
  justify-content: center;
}
```

src/components/MovieCard/ MovieCard.tsx:

```
import './MovieCard.css'

function MovieCard( {movieEl}: any ): JSX.Element {

    const handlePreviewClick = () => {
        window.location.href = "https://www.themoviedb.org/movie/" + movieEl.id;
    }

    return (
        <div className="cardBox">
            <img className="cardBox-poster"
src={"https://image.tmdb.org/t/p/w185" + movieEl.poster_path} alt="poster" />
            <div className="cardBox-content">
                <span className="cardBox-content-title">{movieEl.title}</span>
                <div className="cardBox-content-description">
                    {movieEl.overview}
                </div>
                <button className="cardBox-content-prevButton"
onClick={handlePreviewClick}>Preview</button>
            </div>
        </div>
    );
}

export default MovieCard;
```

src/components/MovieCard/ MovieCard.css:

```
.cardBox {
    display: flex;
    background-color: rgba(255, 255, 255, 0.753);
    border-radius: 10px;
    width: 500px;
    min-height: 300px;
    margin-top: 40px;
    padding: 10px;
}

.cardBox-poster {
    margin-top: 20px;
    border-radius: 10px;
    width: 148px;
    height: 222px;
    margin-left: 20px;
}

.cardBox-content {
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    padding: 15px;
    margin-left: 10px;
}

.cardBox-content-title {
    color: rgb(50, 8, 90);
    font-weight: 600;
    font-size: 20px;
}

.cardBox-content-description {
    margin-top: 10px;
    color: black;
}

.cardBox-content-prevButton {
    font-weight: 700;
    margin-top: 10px;
    width: 150px;
    color: white;
    text-decoration: none;
    padding: .8em 1em calc(.8em + 3px);
    border-radius: 3px;
    background: rgb(64,199,129);
    box-shadow: 0 -3px rgb(53,167,110) inset;
    transition: 0.2s;
}

.cardBox-content-prevButton:hover {
    background: rgb(53, 167, 110);
}
.cardBox-content-prevButton:active {
    background: rgb(33,147,90);
    box-shadow: 0 3px rgb(33,147,90) inset;
}
```

src/components/MovieInfo/ MovieInfo.tsx:

```
import React, { useEffect} from 'react';
import { useHistory, useParams, useRouteMatch } from 'react-router';
import './MovieInfo.css';
import { normalizeUrl } from '../../utils';
import ReturnBtn from '../ReturnBtn/ReturnBtn';
import loading from '../../assets/images/loading.png';
import { useTypedSelector } from '../../hooks/useTypedSelector';
import { ParamsType } from '../../types/types'
import {useMovieActions} from '../../hooks/useMovieActions';
import MovieCard from '../MovieCard/MovieCard';

function MovieInfo() {
    const history: any = useHistory();
    const match = useRouteMatch();
    const params: ParamsType = useParams();
    const cur_url: string = normalizeUrl(match.url);
    const movie = useTypedSelector(state => state.movie.movie);
    const {fetchMovie, clearMovie} = useMovieActions();

    useEffect(() => {
        clearMovie();
        fetchMovie(params.movieName);
    }, [])

    if (movie === 404)
        history.push(`/movieNotFound`);

    return (movie !== null && movie !== 404) ? (
        <div className="container">
            <span className="container-header">Movie Search</span>
            <div className="container-content">
                {
                    movie.map((el: any) => {
                        return (
                            <div className="container-content-cardsBox">
                                <MovieCard movieEl={el} />
                            </div>
                        );
                    })
                }
            </div>
            <ReturnBtn history={history} url={cur_url.slice(0,
cur_url.split('').lastIndexOf('/'))} />
        </div>
    ) : (
        <div className="container">
            <div className="container-header">
                Loading...
            </div>
            <img className="loadingGif" src={loading} alt="loading" />
        </div>
    );
}

export default MovieInfo;
```

src/components/MovieInfo/ MovieInfo.css:

```
.container {
    padding: 0px;
    color: rgb(161, 83, 172);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.container-header {
    font-size: 60px;
    color: rgb(161, 83, 172);
    font-family: Wave-Regular;
}

.container-content {
    display: flex;
    flex-direction: row;
    justify-content: space-evenly;
    flex-wrap: wrap;
    border-radius: 10px;
    padding: 20px;
```

```css
    margin-top: 50px;
    width: 80%;
}

.container-content-mainInfo {
    display: flex;
    width: 100%;
}

.container-content-additionalInfo {
    display: flex;
    width: 100%;
    flex-direction: column;
}

.container-content-additionalInfo-header {
    color: white;
    font-weight: 500;
    margin-top: 20px;
    font-size: 40px;
}

.container-content-additionalInfo-info {
    color: rgb(218, 214, 214);
    font-size: 30px;
    margin-left: 80px;
    margin-top: 5px;
}

.container-content-mainInfo-text {
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    margin-left: 60px;
    margin-top: 10px;
    color: white;
}

.container-content-mainInfo-text-login {
    font-size: 50px;
    text-decoration: none;
    color: rgb(161, 83, 172);
}

.container-content-mainInfo-text-login:hover {
    font-size: 50px;
    text-decoration: none;
    color: white;
}

.container-content-mainInfo-text-name {
    font-size: 30px;
    color:rgb(131, 125, 125);
}

.container-content-mainInfo-avatar {
    margin: 10px;
    width: 40%;
    max-width: 300px;
    min-width: 150px;
    height: 40%;
    border-radius: 50%;
}

.loadingGif {
    width: 100px;
    height: 100px;
    margin-top: 150px;
```

```css
}

@media screen and (max-width: 1000px) {
    .container-content {
        width: 80%;
    }
}

@media screen and (max-width: 600px) {
    .container-content {
        width: 80%;
    }

    .container-content-mainInfo-text {
        margin-left: 40px;
    }

    .container-content-mainInfo-text-login {
        font-size: 40px;
    }

    .container-content-mainInfo-text-login:hover {
        font-size: 40px;
        text-decoration: none;
        color: white;
    }

    .container-content-mainInfo-text-name {
        font-size: 20px;
        color:rgb(131, 125, 125);
    }

    .container-content-additionalInfo-header {
        color: white;
        font-weight: 500;
        margin-top: 20px;
        font-size: 30px;
    }

    .container-content-additionalInfo-info {
        color: rgb(218, 214, 214);
        font-size: 20px;
        margin-left: 80px;
        margin-top: 5px;
    }
}

@media screen and (max-width: 600px) {
    .container-content-mainInfo-avatar {
        min-width: 100px;
    }

    .container-content {
        width: 90%;
    }
}
```

src/components/MovieNotFound/ MovieNotFound.tsx:

```tsx
import './MovieNotFound.css'
import ReturnBtn from '../ReturnBtn/ReturnBtn'
import { useHistory, useRouteMatch } from 'react-router';
import { normalizeUrl } from '../../utils'
import {useMovieActions} from '../../hooks/useMovieActions';
import { useEffect } from 'react';

function MovieNotFound() {
    const history: any = useHistory();
    const match = useRouteMatch();
    const cur_url: string = normalizeUrl(match.url);
    const {clearMovie} = useMovieActions();

    useEffect(() => {
        clearMovie();
    })


    return(
        <div className="page">
            <span className="page-header">Movie Search</span>
            <div className="page-content">
                Movie does not exist =(<br />     Please try again
            </div>
            <ReturnBtn history={history} url={cur_url} />
        </div>
    );
}

export default MovieNotFound;
```

src/components/MovieNotFound/ MovieNotFound.css:

```
@font-face {
    font-family: Wave-Regular;
    src: url(../../assets/fonts/Wave-Regular.ttf);
}

.page {
    padding: 0px;
    color: rgb(161, 83, 172);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.page-header {
    font-size: 60px;
    font-family: Wave-Regular;
}

.page-content {
    margin-top: 100px;
    font-size: 50px;
    color: white;
    text-align: center;
}

@media screen and (max-width: 530px) {
    .page-content {
        margin-top: 100px;
        font-size: 40px;
        color: white;
        text-align: center;
    }
}
```

src/components/PrivateRoute/ index.tsx:

```
import React from 'react';
import { Redirect, Route, RouteComponentProps } from 'react-router-dom';

export const PrivateRoute: React.FC<{ path: string; exact?: boolean,
component?: React.ComponentType<RouteComponentProps<any>> |
React.ComponentType<any>}> = ({
    path,
    component,
    exact
}) => {

    if (!localStorage.getItem('user')) {
        return <Redirect to="/auth" />;
    }
    return (
        <Route exact={exact} path={path} component={component} />
    );
};
```

src/components/ReturnBtn/ReturnBtn.tsx:

```
import {useMovieActions} from '../../hooks/useMovieActions';
import './ReturnBtn.css'
import {useHistory} from "react-router";

export default function ReturnBtn( {url}: any ): JSX.Element {
    const history = useHistory();
    const cur_url: string = url.slice(0, url.split('').lastIndexOf('/'));
    const {clearMovie} = useMovieActions();

    const goBack = () => {
        history.push(`${cur_url}/`);
        clearMovie()
    }

    return (
    <button id="return-btn" className="returnBtn" onClick={goBack}>
        Go Back
    </button>
    );
}
```

src/components/ReturnBtn/ReturnBtn.css:

```
.returnBtn {
    margin-top: 30px;
    width: 150px;
    height: 50px;
    color:rgb(161, 83, 172);
    font-size: 30px;
    background-color: rgb(32, 36, 39);
}
```

src/components/SearchPage/ SearchPage.tsx:

```
import React, {RefObject, useCallback, useState} from 'react'
import {useRouteMatch, useHistory} from 'react-router-dom';
import './SearchPage.css'
import { normalizeUrl} from '../../utils'
import {useMovieActions} from '../../hooks/useMovieActions';
import { useTypedSelector } from '../../hooks/useTypedSelector'
import SearchSuggests from '../SearchSuggests/SearchSuggests';
import axios from "axios";
```

```
function SearchPage(): JSX.Element {
    const history: any = useHistory();
    const match = useRouteMatch();
    const input = React.useRef<HTMLInputElement>(null);
    const inputDelete = React.useRef<HTMLInputElement>(null);
    const cur_url: string = normalizeUrl(match.url);
    const suggestsBoxRef: any = React.useRef<HTMLDivElement>(null);
    const suggestsBoxDeleteRef: any = React.useRef<HTMLDivElement>();
    const {fetchMovie} = useMovieActions();
    const movie = useTypedSelector(state => state.movie.movie);
    const [movieToDelete, setMovieToDelete] = useState<string>('');

    const deleteData = useCallback(() => {
        axios.delete(`http://localhost:3001/movies/${movieToDelete}`)
            .then(() => alert('success'))
            .catch(() => alert('error'))
    }, [movieToDelete])

    const putData = () => {
        axios.put('http://localhost:3001/movies/2', {
            id: 2,
            name: 'Доктор кто?'
        })
    }

    const postData = () => {
        axios.post('http://localhost:3001/movies/',
[{"adult":false,"backdrop_path":null,"genre_ids":[878,28,12],"id":700330,"origin
al_language":"en","original_title":"Untitled Taika Waititi Star Wars
Film","overview":"","popularity":1.4,"poster_path":null,"title":"Звёздные
войны","video":false,"vote_average":0,"vote_count":0},{"adult":false,"backdrop_p
ath":null,"genre_ids":[878,16,35],"id":42982,"original_language":"en","original_
title":"Robot Chicken: Star Wars Episode II","overview":"Второй эпизод из серии
пародий на сагу о Звездных
Войнах.","popularity":7.534,"poster_path":"/15L0z57jELdzynqcbWq0fFFou48.jpg","re
lease_date":"2008-11-16","title":"Робоцып: Звездные войны. Эпизод
II","video":false,"vote_average":7.5,"vote_count":101},{"adult":false,"backdrop_
path":null,"genre_ids":[35,878,16],"id":51888,"original_language":"en","original
_title":"Robot Chicken: Star Wars Episode III","overview":"Третий эпизод из
серии пародий на сагу о Звездных
Войнах.","popularity":7.681,"poster_path":"/mi2lVho2zpfwcxI6yC1QYJi435D.jpg","re
lease_date":"2010-12-19","title":"Робоцып: Звездные войны. Эпизод
III","video":false,"vote_average":7.4,"vote_count":105},{"adult":false,"backdrop
_path":"/sNNFLEcAuy4C3RyXCnKoArn7Aty.jpg","genre_ids":[16,28,878,12],"id":12180,
"original_language":"en","original_title":"Star Wars: The Clone
Wars","overview":"На передовых позициях межгалактической войны Добра и Зла —
снова любимые герои: Анакин Скайуокер, Оби-Ван Кеноби и Падме Амидала, а также
присоединившийся к ним падаван Анакина, Асока. Противостоящие им злодеи —
Палпатин, граф Дуку и генерал Гривус — вынашивают план захвата Галактики.
Начинается последняя схватка, и судьбы Вселенной вверены рыцарям-джедаям. Их
подвиги станут самыми яркими эпизодами масштабных битв и подарят множество
удивительных
открытий.","popularity":29.523,"poster_path":"/ywRtBu88SLAkNxD0GFib6qsFkBK.jpg",
"release_date":"2008-08-05","title":"Звёздные войны: Войны
клонов","video":false,"vote_average":6.1,"vote_count":1421},{"adult":false,"back
drop_path":null,"genre_ids":[99],"id":378386,"original_language":"en","original_
title":"Star Wars: Greatest Moments","overview":"Алекс Зейн подсчитывает 20
лучших моментов из «Звездных войн», как проголосовала общественность. Включает в
себя участие известных поклонников, а также звезд и экипажа межгалактической
саги.","popularity":4.272,"poster_path":"/zIffPwISrW48qSmvAXEV27lBTMA.jpg","rele
ase_date":"2015-12-26","title":"Звездные войны: величайшие
моменты","video":false,"vote_average":6.6,"vote_count":19},{"adult":false,"backd
rop_path":"/jn52me8AagfNt7r84SgQbV0R9ZG.jpg","genre_ids":[28,12,878],"id":181812
,"original_language":"en","original_title":"Star Wars: The Rise of
Skywalker","overview":"Фильм завершает невероятную историю семьи Скайуокеров,
длящуюся уже более сорока лет, и обещает дать ответы на все загадки
```

```
из предыдущих серий. Зрителя ожидают старые и новые герои, уникальные миры,
увлекательные путешествия на край Галактики и грандиозный финал фантастической
саги.","popularity":124.076,"poster_path":"/mCD1ugUVWwa1pOmOACQY8sV7eu9.jpg","re
lease_date":"2019-12-18","title":"Звёздные войны: Эпизод 9 - Скайуокер.
Восход","video":false,"vote_average":6.5,"vote_count":6991},{"adult":false,"back
drop_path":"/uNjBnOmdjZoiWTLQ938YJZ1cYVU.jpg","genre_ids":[16,35,10751,878,12,10
770],"id":392216,"original_language":"en","original_title":"Phineas and Ferb:
Star
Wars","overview":"","popularity":10.487,"poster_path":"/xomphpz7MIasqVluPX83TjoT
L8G.jpg","release_date":"2014-07-26","title":"Финес и Ферб: Звездные
войны","video":false,"vote_average":7,"vote_count":116},{"adult":false,"backdrop
_path":"/1Lhc32P0a62BgMFgd20wXR1osR3.jpg","genre_ids":[16,12,35,10751,878],"id":
732670,"original_language":"en","original_title":"The Lego Star Wars Holiday
Special","overview":"В спецвыпуске Рей, Финн, По, Чуи, Роуз и дроиды
воссоединятся на экране для празднования Дня жизни. Чтобы лучше познать силу,
Рей вместе с BB-8 прилетает в таинственный храм джедаев, откуда с помощью
загадочного артефакта отправляется в приключение по любимым моментам франшизы.
Она встретится с Люком Скайуокером, Дартом Вейдером, Йодой, Оби-Ваном и другими
знаковыми героями и
злодеями.","popularity":47.04,"poster_path":"/zyzJSI7UZZzz5Toj10rYGF5689z.jpg","
release_date":"2020-11-17","title":"ЛЕГО Звёздные войны: Праздничный
спецвыпуск","video":false,"vote_average":6.7,"vote_count":187},{"adult":false,"b
ackdrop_path":"/jTB8BKStGgIktED2yVMA6U6mppl.jpg","genre_ids":[16,878,35,10751],"
id":79158,"original_language":"en","original_title":"LEGO Star Wars: Bombad
Bounty","overview":"","popularity":2.402,"poster_path":"/r088BCbLiSxccTHXxIZBZ5o
lRvM.jpg","release_date":"2010-11-27","title":"Lego Звездные войны: Награда
Бомбада","video":false,"vote_average":5.8,"vote_count":19},{"adult":false,"backd
rop_path":null,"genre_ids":[16,35,878],"id":42979,"original_language":"en","orig
inal_title":"Robot Chicken: Star Wars","overview":"«Робоцып» — калейдоскоп
коротких скетчей, полных чёрного юмора и сатиры. Практически каждая сценка
пародирует кино, теле-шоу, компьютерные игры, героев мультфильмов и комиксов,
актёров, политиков, певцов и других знаменитостей, а также продукты питания,
рекламные ролики или прочие проявления массовой культуры. На этот раз, это
«Звездные
Войны»…","popularity":7.968,"poster_path":"/h44WN4mVJ6wEpJgLaaNoFjv0NAo.jpg","re
lease_date":"2007-07-17","title":"Робоцып: Звездные
войны","video":false,"vote_average":7.2,"vote_count":176},{"adult":false,"backdr
op_path":null,"genre_ids":[16,878,10751],"id":81233,"original_language":"en","or
iginal_title":"LEGO Star Wars: The Quest for R2-D2","overview":"В новом
измерении звёздных войн, построенных из конструктора LEGO, всё маленькое, милое
и забавное. Анакин Скайуокер и R2-D2 на истребителе спасаются с вражеской
планеты, где они завершили важную миссию. При атаке преследователей R2-D2
выпадает с корабля и на парашуте приземляется в пустыню, где попадает в
плен…","popularity":1.568,"poster_path":"/pfI6dMB5aplRYe7vSniDznnFWUx.jpg","rele
ase_date":"2009-06-27","title":"Lego Звездные войны: Поиск R2-
D2","video":false,"vote_average":5.2,"vote_count":20},{"adult":false,"backdrop_p
ath":"/ae9xlnkS2qb5Dy9Mtlu68AWh42O.jpg","genre_ids":[12,35,10751,878,10770],"id"
:74849,"original_language":"en","original_title":"The Star Wars Holiday
Special","overview":"","popularity":9.348,"poster_path":"/1TY4OAkcHRovlHDxSLW7UD
Jlild.jpg","release_date":"1978-12-01","title":"Звездные войны: Праздничный
спецвыпуск","video":false,"vote_average":3.2,"vote_count":326},{"adult":false,"b
ackdrop_path":"/A0wfk0gISF7gH78CKCfLlkAf1Vs.jpg","genre_ids":[878,12,28],"id":18
95,"original_language":"en","original_title":"Star Wars: Episode III - Revenge
of the Sith","overview":"Идёт третий год Войн клонов. Галактическая Республика,
некогда бывшая спокойным и гармоничным государством, превратилась в поле битвы
между армиями клонов, возглавляемых канцлером Палпатином, и армадами дроидов,
которых ведёт граф Дуку, тёмный лорд ситхов. Республика медленно погружается во
тьму. Лишь рыцари-джедаи, защитники мира и справедливости, могут противостоять
злу, которое вскоре поглотит галактику. Но настоящая битва идёт в душе у
молодого рыцаря-джедая Энакина, который разрывается между долгом джедая и
любовью к своей жене, сенатору Падме Амидале. И от того, какое чувство в нём
победит, зависит будущее всего
мира.","popularity":25.63,"poster_path":"/yshcD6BJ66tHvLKyKR7BcSQo3MQ.jpg","rele
ase_date":"2005-05-17","title":"Звёздные войны: Эпизод 3 - Месть
Ситхов","video":false,"vote_average":7.4,"vote_count":10593},{"adult":false,"bac
kdrop_path":"/hVl0qfLZ5loznDgQ0HxCTZqvZhz.jpg","genre_ids":[16,10751,878,35,12],
"id":70608,"original_language":"en","original_title":"LEGO Star Wars: The
Padawan Menace","overview":"Рядовая экскурсия Академии джедаев неожиданно
```

```
превращается в веселое приключение в мультфильме Lego. Звездные войны:
Падаванская угроза. Мастер Йода сопровождает группу джедаев-экскурсантов в
палаты Сената, как вдруг ощущает возмущение Силы. Он призван помочь спасти
Республику, но, оказавшись на своем корабле, обнаруживает, что юный джедай Йен
тайком проник туда... и теперь жаждет приключений. Тем временем C-3PO и R2-D2,
оставшиеся присматривать за группой экскурсантов, понимают, что влипли по уши. И
пока коварные ситхи готовятся нанести решающий удар, Йода и Дроиды должны
позаботиться, чтобы их подопечные не рассыпались на
кубики.","popularity":11.269,"poster_path":"/dzI7fwoZHi9uquYZqTh0WVxpnOq.jpg","r
elease_date":"2011-07-22","title":"LEGO Звездные войны: Падаванская
угроза","video":false,"vote_average":6.3,"vote_count":98},{"adult":false,"backdr
op_path":"/zqkmTXzjkAgXmEWLRsY4UpTWCeo.jpg","genre_ids":[12,28,878],"id":11,"ori
ginal_language":"en","original_title":"Star Wars","overview":"Татуин. Планета-
пустыня. Уже постаревший рыцарь Джедай Оби Ван Кеноби спасает молодого Люка
Скайуокера, когда тот пытается отыскать пропавшего дроида. С этого момента Люк
осознает свое истинное назначение: он один из рыцарей Джедай. В то время как
гражданская война охватила галактику, а войска повстанцев ведут бои против сил
злого Императора, к Люку и Оби Вану присоединяется отчаянный пилот-наемник Хан
Соло, и в сопровождении двух дроидов, R2D2 и C-3PO, этот необычный отряд
отправляется на поиски предводителя повстанцев — принцессы
Леи.","popularity":58.915,"poster_path":"/291yZULc4NBE9QDxqHGAJUk41j7.jpg","rele
ase_date":"1977-05-25","title":"Звёздные войны: Эпизод 4 - Новая
надежда","video":false,"vote_average":8.2,"vote_count":15906},{"adult":false,"ba
ckdrop_path":"/5fu7fzy4NZTsL1Jap00UBIInAuB.jpg","genre_ids":[12,28,878],"id":189
3,"original_language":"en","original_title":"Star Wars: Episode I - The Phantom
Menace","overview":"Мирная и процветающая планета Набу. Торговая федерация,
не желая платить налоги, вступает в прямой конфликт с королевой Амидалой,
правящей на планете, что приводит к войне. На стороне королевы и республики
в ней участвуют два рыцаря-джедая: учитель и ученик, Квай-Гон-Джин и Оби-Ван
Кеноби...","popularity":28.619,"poster_path":"/l1LyrhQwDGur18rxRXKQJQhvx7Y.jpg",
"release_date":"1999-05-19","title":"Звёздные войны: Эпизод 1 - Скрытая
угроза","video":false,"vote_average":6.5,"vote_count":11310},{"adult":false,"bac
kdrop_path":"/dNt5q68BBkddBxlvrHDa1apyBy8.jpg","genre_ids":[12,28,878],"id":1894
,"original_language":"en","original_title":"Star Wars: Episode II - Attack of
the Clones","overview":"Действие разворачивается через 10 лет после событий,
описанных в первом эпизоде знаменитой саги. Республика все глубже погружается
в пучину противоречий и хаоса. Движение сепаратистов, представленное сотнями
планет и могущественным альянсом корпораций, грозит стать новой угрозой для
Галактики, с которой не смогут справиться даже
джедаи.","popularity":25.105,"poster_path":"/jGiIdLMD5Y16snttJwUcQSDBXUv.jpg","r
elease_date":"2002-05-15","title":"Звёздные войны: Эпизод 2 - Атака
клонов","video":false,"vote_average":6.5,"vote_count":10218},{"adult":false,"bac
kdrop_path":"/aM1dP91YFVPsgq1WG8vMwip9zLe.jpg","genre_ids":[878,16,10751,35,12],
"id":136406,"original_language":"en","original_title":"LEGO Star Wars: The
Empire Strikes Out","overview":"Во времена гражданской войны Галактическая
империя угрожала подавить народное восстание с помощью совершенного оружия:
Звезды смерти. Принцесса Лиа и повстанческий альянс одержали важную победу, но
праздновать некогда. Чтобы избежать преследования за уничтожение Звезды смерти
они принимают решение эвакуироваться на секретную базу, которая находится на
Хоте…","popularity":4.854,"poster_path":"/ikUhOSuKOd9Sjf6dVP585lFtiLb.jpg","rele
ase_date":"2012-09-24","title":"LEGO Звездные войны: Империя наносит
удар","video":false,"vote_average":6.5,"vote_count":60},{"adult":false,"backdrop
_path":"/k6EOrckWFuz7I4z4wiRwz8zsj4H.jpg","genre_ids":[28,12,878,14],"id":140607
,"original_language":"en","original_title":"Star Wars: The Force
Awakens","overview":"Через тридцать лет после гибели Дарта Вейдера и Императора
галактика по-прежнему в опасности. Государственное образование Новый Порядок
во главе с их таинственным верховным лидером Сноуком и его правой рукой Кайло
Реном идёт по стопам Империи, пытаясь захватить всю власть. В это нелёгкое время
судьба сводит юную девушку Рей и бывшего штурмовика Нового Порядка Финна
с героями времён войны с Империей — Ханом Соло, Чубаккой и Королевой Леей.
Вместе они должны дать бой Новому Порядку, однако настаёт тот момент, когда
становится очевидно, что лишь джедаи могут остановить Сноука и Кайло Рена.
И в галактике в живых остаётся только
один.","popularity":48.946,"poster_path":"/5ltRJxurOTIaMQwcOwSau7l3GME.jpg","rel
ease_date":"2015-12-15","title":"Звёздные войны: Эпизод 7 - Пробуждение
силы","video":false,"vote_average":7.3,"vote_count":16080},{"adult":false,"backd
rop_path":"/iP2tEA2A77qhbhRfcFkiD6WFOqH.jpg","genre_ids":[12,28,878],"id":1892,"
original_language":"en","original_title":"Return of the
```

```
Jedi","overview":"В заключительном шестом эпизоде «Звездных войн» Дарт Вейдер
создает вторую «Звезду Смерти». Он объединяет все силы зла, чтобы с помощью
этого смертоносного оружия нанести последний сокрушительный удар по повстанцам
и их союзникам. Люк Скайуокер вместе с принцессой Лейей и верными дроидами R2D2
и C-3PO отправляется спасать своего друга Хана Соло, который попал в плен
к отвратительному Джаббе Хатту — могущественному повелителю
преступников.","popularity":23.34,"poster_path":"/qK7EmIccOeL16jzhwpvNzBSOQxS.jp
g","release_date":"1983-05-25","title":"Звёздные войны: Эпизод 6 - Возвращение
Джедая","video":false,"vote_average":7.9,"vote_count":11743}])
    }

    const findMovie = () => {
        if (input.current !== null && input.current.value.trim().length > 0)
            history.push(`${cur_url}/movieInfo/${input.current.value.trim()}`);
    };
    const enterPress = (e: any) => {
        if(e.key === 'Enter')
            findMovie();
    };

    const handleChangeSearch = (e: any) => {
        const search_str: string = e.target.value.trim();
        if (search_str.length > 0)
            fetchMovie(search_str);
    }

    const handleInputClick = (ref: RefObject<any>) => {
        if (ref !== null)
                ref.current.style.display = 'flex';
    }

    const handleDeleteInputChange = useCallback((value) => {
        const search_str: string = value.trim();
        if (search_str.length > 0)
            fetchMovie(search_str);
        setMovieToDelete(value);
    }, [setMovieToDelete])

    return (
        <div className="finder">
            <span className="finder-header">Movie Search</span>
            <div className="finderBox">
                <span className="finderBox-text">Enter your request</span>
                <div className="finderBox-searchFieldBox">
                    <div className="finderBox-searchFieldBox-content">
                        <input
                            onClick={() => handleInputClick(suggestsBoxRef)}
                            className="finderBox-input"
                            placeholder="Type to search"
                            type="text"
                            ref={input}
                            onKeyPress={enterPress}
                            onChange={handleChangeSearch}
                        />
                        <div className="suggestsBox" ref={suggestsBoxRef}>
                            {
                                movie && movie !== 404 && (
                                    <SearchSuggests input={input}
suggestsBox={suggestsBoxRef} />
                                )
                            }
                        </div>
                    </div>
                    <button className="finderBox-btn"
onClick={findMovie}>Find</button>
                </div>
                <div className="finderBox-searchFieldBox">
                    <div className="finderBox-searchFieldBox-content">
```

```
                        <input
                            className="finderBox-input"
                            placeholder="Enter title"
                            type="text"
                            value={movieToDelete}
                            onChange={(e) =>
handleDeleteInputChange(e.target.value)}
                            ref={inputDelete}
                            onClick={() =>
handleInputClick(suggestsBoxDeleteRef)}
                        />
                        <div className="suggestsBox" ref={suggestsBoxDeleteRef}>
                            {
                                movie && movie !== 404 && (
                                    <SearchSuggests input={inputDelete}
suggestsBox={suggestsBoxDeleteRef} />
                                )
                            }
                        </div>
                    </div>
                    <button className="finderBox-btn"
onClick={deleteData}>Delete</button>
                </div>
                <button className="finderBox-btn"
onClick={postData}>Post</button>
                <button className="finderBox-btn" onClick={putData}>Put</button>
            </div>
        </div>
    )
}

export default SearchPage;
```

src/components/SearchPage/ SearchPage.css:

```
@font-face {
    font-family: Wave-Regular;
    src: url(../../assets/fonts/Wave-Regular.ttf);
}

.finder {
    color: rgb(161, 83, 172);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.finderBox {
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: flex-start;
    margin: 100px 0px 0px 0px;
    border-radius: 10px;
    padding: 10px;
    width: 60%;
    min-height: 200px;
}

.finderBox-text {
    font-size: 30px;
    font-family: sans-serif;
}

.finderBox-searchFieldBox-content {
```

```css
    width: 60%;
}

.finderBox-input {
    margin-top: 20px;
    padding-left: 20px;
    border: none;
    outline: none;
    border: 1px solid white;
    border-radius: 5px;
    height: 40px;
    width: 90%;
    background-color: rgb(19, 23, 27);
    font-size: 20px;
    color: white;
}

.finderBox-input:hover, .finderBox-input:focus {
    border: 1px solid rgb(76, 89, 145);
}
.finderBox-input:focus {
    background-color: bisque;
    color: black;
}

.finderBox-searchFieldBox {
    display: flex;
    flex-direction: row;
    width: 100%;
    justify-content: center;
}

.finderBox-btn {
    margin-top: 20px;
    margin-left: 20px;
    width: 150px;
    height: 50px;
    color:rgb(161, 83, 172);
    font-size: 30px;
    background-color: rgb(32, 36, 39);
}

.finder-header {
    font-size: 60px;
    font-family: Wave-Regular;
    text-align: center;
}

@media screen and (max-width: 500px) {
    .finderBox-text {
        font-size: 25px;
    }
    .finderBox {
        width: 80%;
    }
    .finderBox-input {
        width: 80%;
    }
}
```

src/components/SearchSuggests/ SearchSuggests.tsx:

```
import {useTypedSelector} from '../../hooks/useTypedSelector'
import { useEffect } from 'react'
import { handleTextHighlight } from '../../utils'
import './SearchSuggests.css'

function SearchSuggests({ input, suggestsBox }: any): JSX.Element {
    const movie = useTypedSelector(state => state.movie.movie);

    const handleSuggestClick = (e: any) => {
        input.current.value = e.target.id;
    }

    useEffect(() => {
        document.addEventListener('click', (e) => {
            if (suggestsBox !== null && suggestsBox.current !== null && e.target
!== input.current)
                suggestsBox.current.style.display = 'none';
        })
    }, [input, suggestsBox])

    return (
        <div>
            {
                movie && movie.length && (
                    movie.map((el: any) => {
                        const decomposed_str = handleTextHighlight(el.title,
input);
                        return (
                            <button className="suggestsBox-suggest"
onClick={handleSuggestClick} id={decomposed_str.before +
decomposed_str.search_str + decomposed_str.after}>
                                <span id={decomposed_str.before +
decomposed_str.search_str + decomposed_str.after}>{decomposed_str.before}</span>
                                <span className="suggestsBox-suggest-search_str"
id={decomposed_str.before + decomposed_str.search_str +
decomposed_str.after}>{decomposed_str.search_str}</span>
                                <span id={decomposed_str.before +
decomposed_str.search_str + decomposed_str.after}>{decomposed_str.after}</span>
                            </button>
                        )
                    })
                )
            }
        </div>
    )
}

export default SearchSuggests;
```

src/components/SearchSuggests/ SearchSuggests.css:

```
.suggestsBox {
    display: none;
    flex-direction: column;
    border-radius: 10px;
    background-color: white;
    width: 90%;
    position: absolute;
}

.suggestsBox-suggest {
    padding-left: 15px;
    width: 100%;
    margin: 0;
    height: 40px;
    border: 0;
    background-color: white;
    text-align: start;
    border-radius: 10px;
    font-size: 15px;
}

.suggestsBox-suggest:hover {
    background-color: rgb(179, 172, 172);
}

.suggestsBox-suggest-search_str {
    color: rgb(161, 83, 172);
    font-weight: 700;
}
```

Тесты:

movie.test.ts:

```
import {MovieActionType, MovieState} from '../types/types'
import {movieReducer} from "../store/reducers/movie";

const defaultState: MovieState = {
    movie: [],
}

describe('movie reducer test', () => {
    it('new movie should be added', () => {
        const payload = {
            title: 'Шрек',
            url: 'link'
        };
        let newState = movieReducer(defaultState, {
            type: MovieActionType.ADD_MOVIE,
            payload: payload
        });
        expect(newState.movie[0].title).toBe('Шрек');
    })
    it('movies should be deleted', () => {
        const initState = {
            movie: [
                {
                    title: 'Шрек 2',
                    url: 'link'
                },
                {
                    title: 'Шрек 2',
                    url: 'link'
                },
                {
                    title: 'Шрек 2',
                    url: 'link'
                },
                {
                    title: 'Шрек 2',
                    url: 'link'
                },
            ]
        }
        let newState = movieReducer(initState, {
            type: MovieActionType.CLEAR_MOVIE,
            payload: []
        });
        expect(newState.movie.length).toBe(0);
    })
})
```

selenium.test.js:

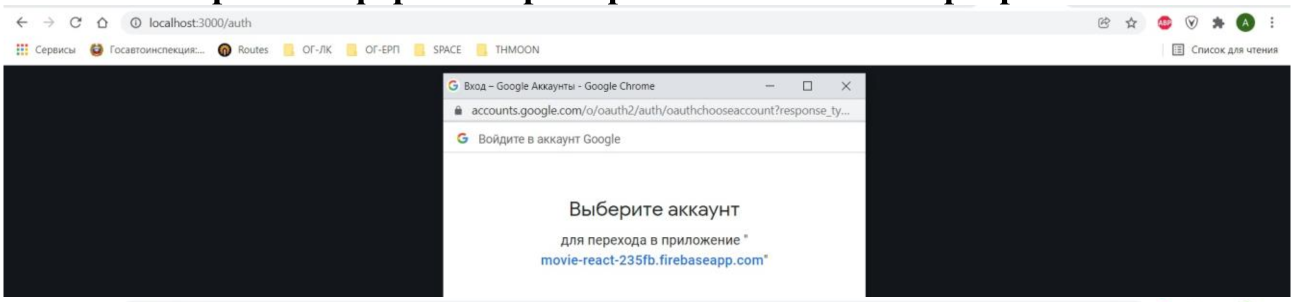
```
const {Builder, By, Key, until} = require('selenium-webdriver');

describe('integration test with selenium chromedriver', () => {
    // let d = new Builder().forBrowser('chrome').build();
    //
    // let chrome_driver;
    //
    // it('waits for the driver to start', () => {
    //     return d.then(_d => {
    //         chrome_driver = _d;
    //     })
    // })

    it('items list should not be empty', async () => {
        let driver = await new Builder().forBrowser('chrome').build();
        await driver.get('http://localhost:3000/auth');
        driver.executeScript(() => {
            localStorage.setItem('user', 'test');
        })
        await driver.sleep(500);
        await driver.get('http://localhost:3000/');
        let finderBox = await driver.findElement(By.className('finderBox-
input'));
        finderBox.sendKeys('Звёздные');
        let findBtn = await driver.findElement(By.className('finderBox-btn'));
        findBtn.sendKeys(Key.ENTER);
        let list = await driver.wait(
            until.elementsLocated(By.className('cardBox')),
            20000
        );
        // if (login)
        //     console.log(login)
        // else
        //     throw new Error('Error')
        // driver.getAllWindowHandles().then((handles) => {
        //     driver.switchTo().window(handles[1]).then(() => {
        //         let loginInput = driver.wait(
        //             until.elementLocated(By.name('identifier')),
        //             2000
        //         )
        //         loginInput.sendKeys('gmail.com');
        //     })
        // })
        expect(list.length).toBeGreaterThanOrEqual(0);
        driver.quit();
    })
})
```

snapshot.test.js:

```
import ReturnBtn from "../components/ReturnBtn/ReturnBtn";
import {Provider} from "react-redux";
import {store} from "../store/index";
import {render} from "enzyme";
import {List} from "../components/Tests/List";

describe('ReturnBtn test', () => {
    it('should render correctly', () => {
        const output = render(
            <Provider store={store}>
                <ReturnBtn url="mockUrl" />
            </Provider>
        );
        expect(output).toMatchSnapshot();
    });
});
```

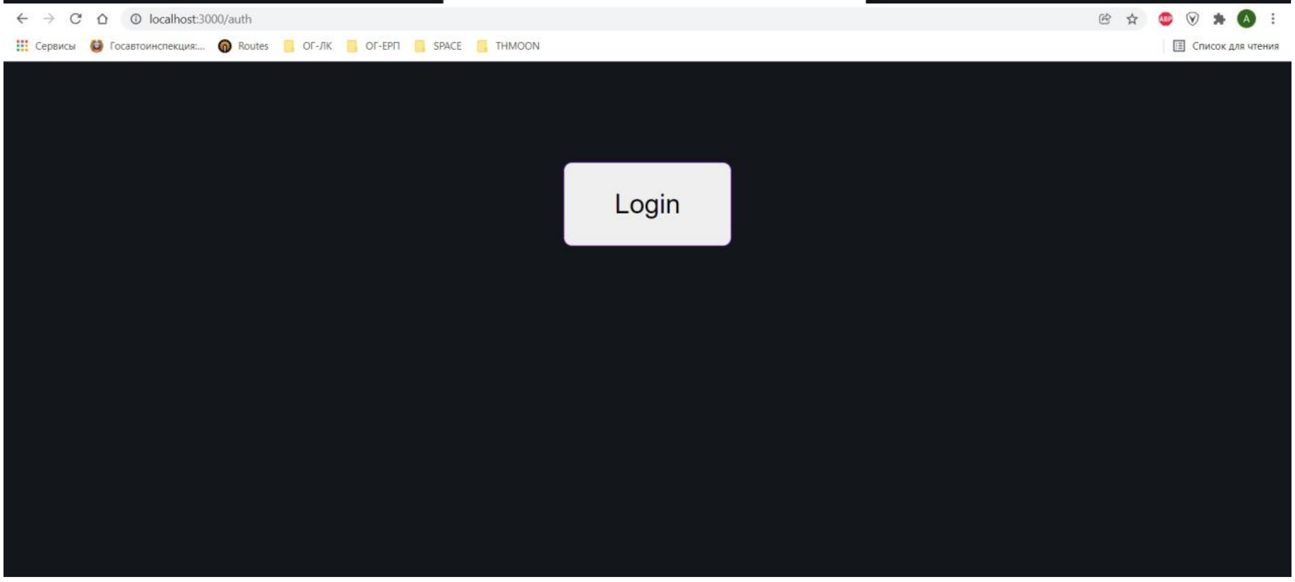
```
describe('List test', () => {
    it('filled list snapshot', () => {
        const list = render(<List items={['lorem', 'lorem', 'lorem']} />);

        expect(list).toMatchSnapshot();
    })

    it('empty list snapshot', () => {
        const list = render(<List />);

        expect(list).toMatchSnapshot();
    })
})
```

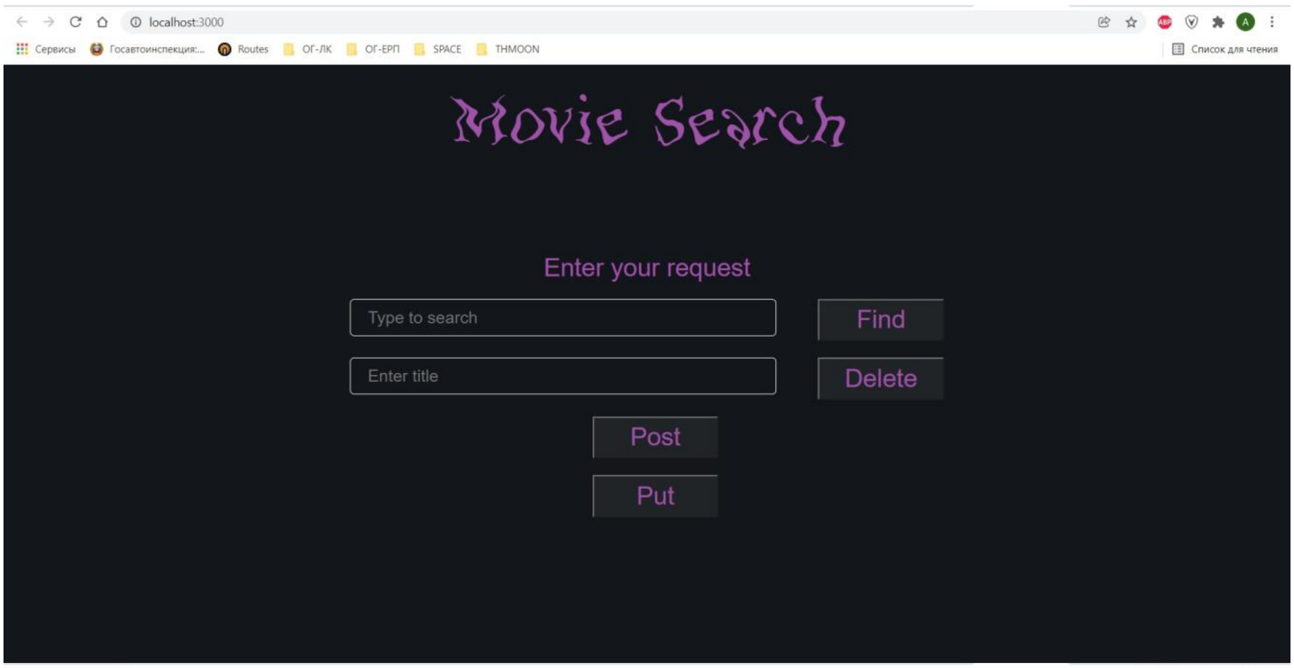
Экранные формы с примерами выполнения программы:



Выберите аккаунт

для перехода в приложение "movie-react-235fb.firebaseapp.com"





Movie Search

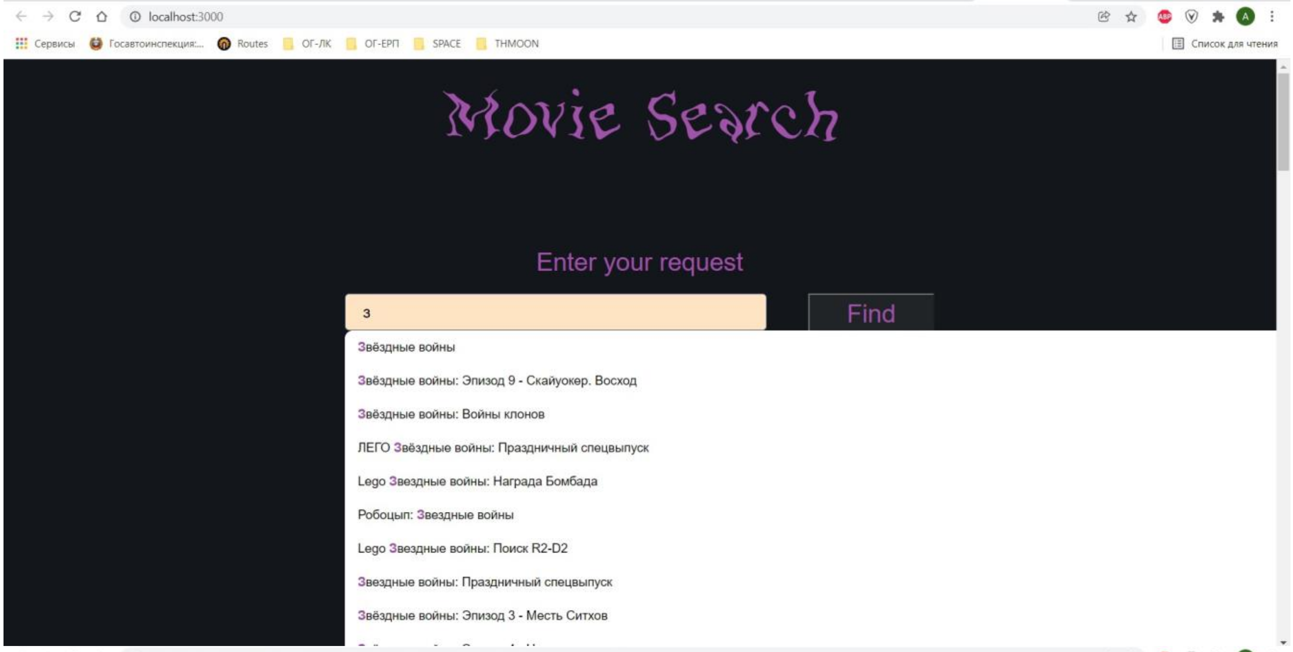
Enter your request

Type to search | Find

Enter title | Delete

Post

Put



Movie Search

Enter your request

з | Find

Звёздные войны

Звёздные войны: Эпизод 9 - Скайуокер. Восход

Звёздные войны: Войны клонов

ЛЕГО Звёздные войны: Праздничный спецвыпуск

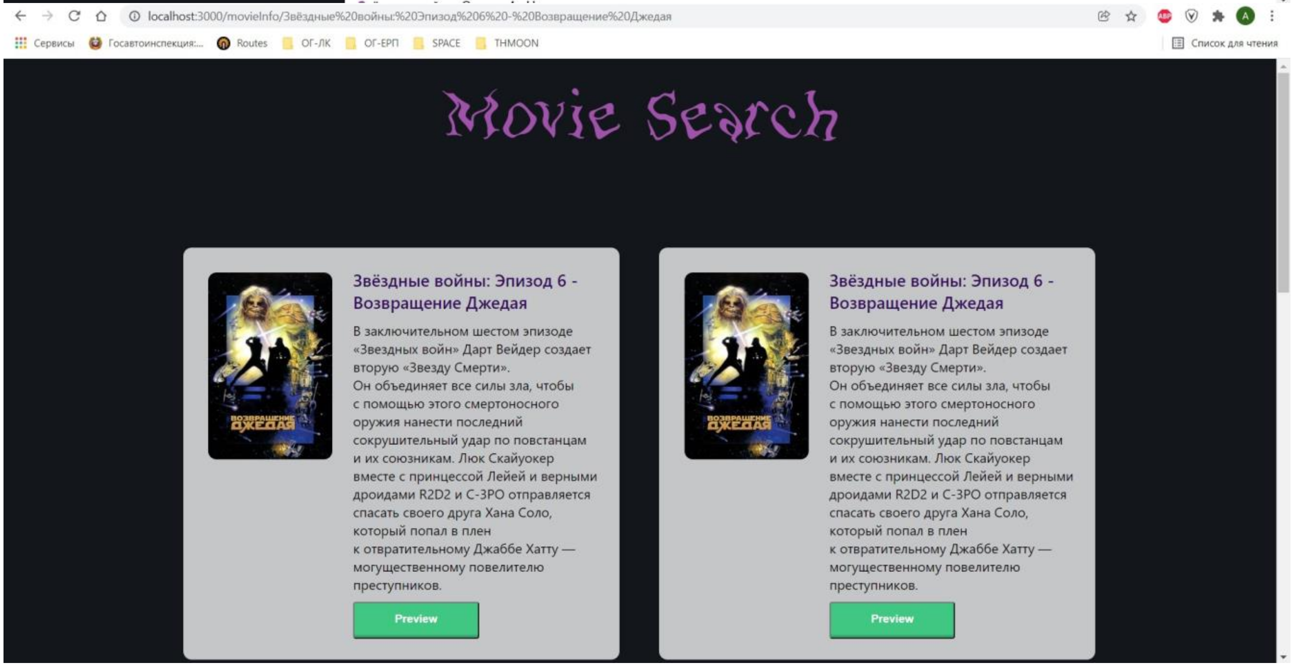
Lego Звездные войны: Награда Бомбада

Робоцып: Звездные войны

Lego Звездные войны: Поиск R2-D2

Звездные войны: Праздничный спецвыпуск

Звёздные войны: Эпизод 3 - Месть Ситхов


localhost:3000/movieInfo/Звёздные%20войны:%20Эпизод%206%20-%20Возвращение%20Джедая
Сервисы
Госавтоинспекция:...
Routes
ОГ-ЛК
ОГ-ЕРП
SPACE
THMOON
Список для чтения

Movie Search

Звёздные войны: Эпизод 6 - Возвращение Джедая

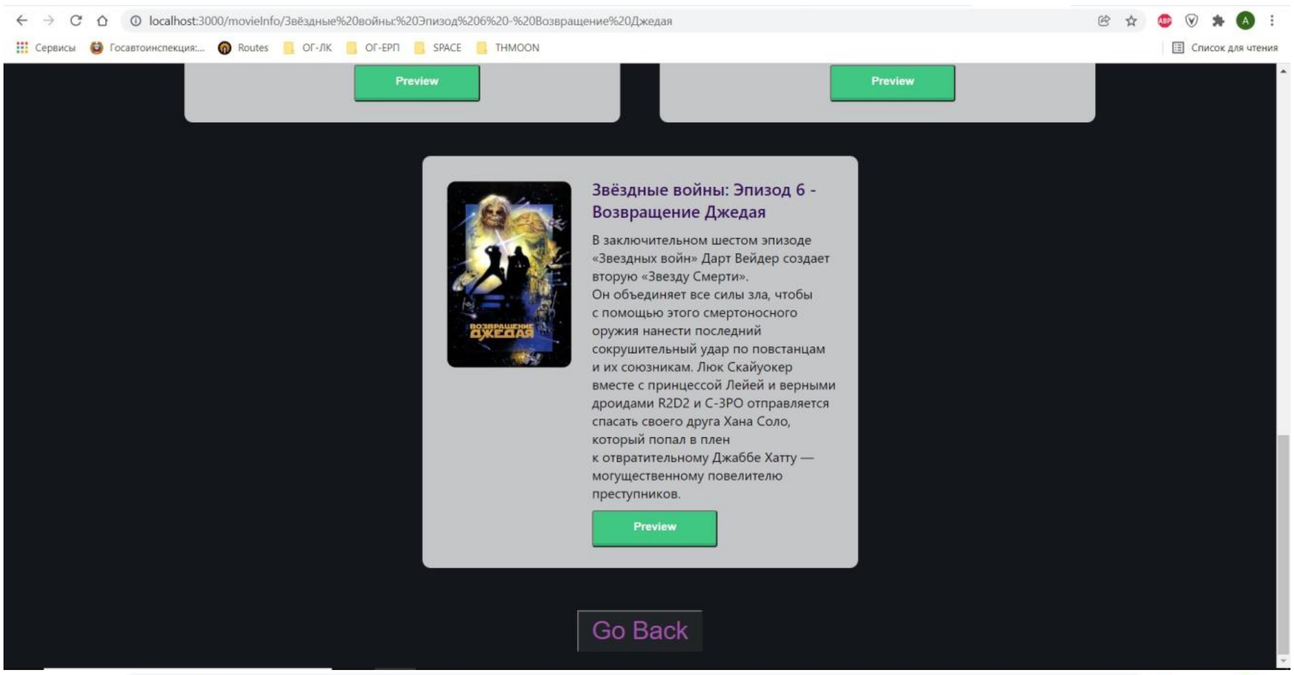
В заключительном шестом эпизоде «Звездных войн» Дарт Вейдер создает вторую «Звезду Смерти».
Он объединяет все силы зла, чтобы с помощью этого смертоносного оружия нанести последний сокрушительный удар по повстанцам и их союзникам. Люк Скайуокер вместе с принцессой Лейей и верными дроидами R2D2 и C-3PO отправляется спасать своего друга Хана Соло, который попал в плен к отвратительному Джаббе Хатту — могущественному повелителю преступников.

Preview

Звёздные войны: Эпизод 6 - Возвращение Джедая

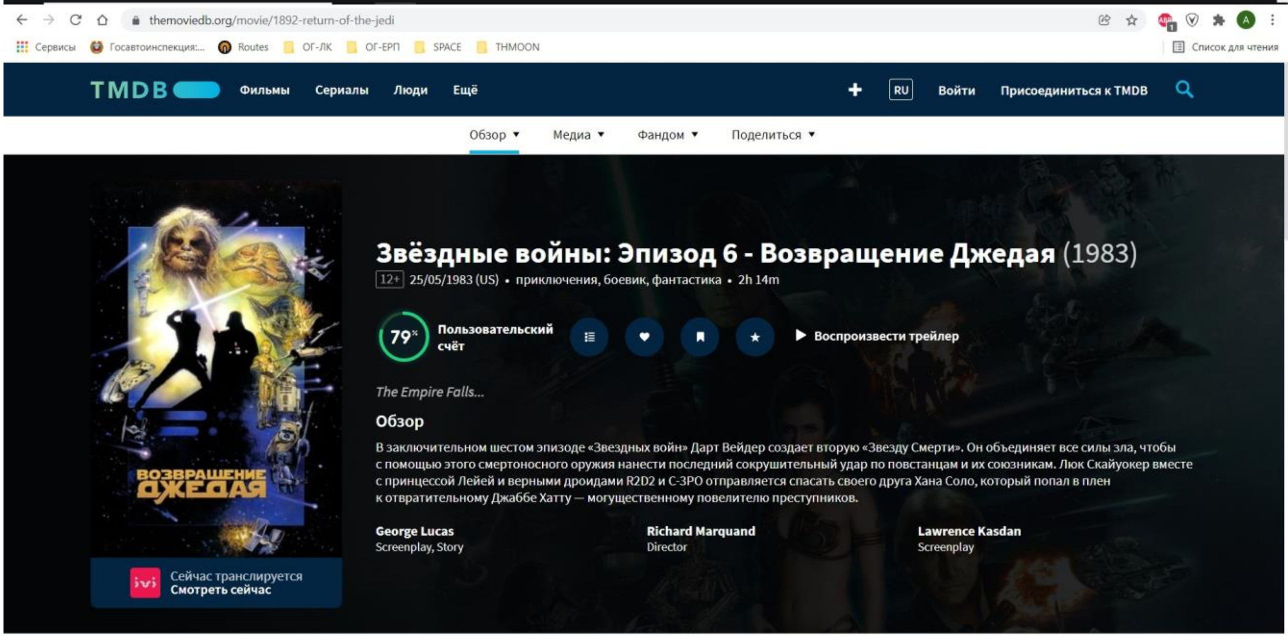
В заключительном шестом эпизоде «Звездных войн» Дарт Вейдер создает вторую «Звезду Смерти».
Он объединяет все силы зла, чтобы с помощью этого смертоносного оружия нанести последний сокрушительный удар по повстанцам и их союзникам. Люк Скайуокер вместе с принцессой Лейей и верными дроидами R2D2 и C-3PO отправляется спасать своего друга Хана Соло, который попал в плен к отвратительному Джаббе Хатту — могущественному повелителю преступников.

Preview



Звёздные войны: Эпизод 6 - Возвращение Джедая

В заключительном шестом эпизоде «Звездных войн» Дарт Вейдер создает вторую «Звезду Смерти». Он объединяет все силы зла, чтобы с помощью этого смертоносного оружия нанести последний сокрушительный удар по повстанцам и их союзникам. Люк Скайуокер вместе с принцессой Лейей и верными дроидами R2D2 и C-3PO отправляется спасать своего друга Хана Соло, который попал в плен к отвратительному Джаббе Хатту — могущественному повелителю преступников.



Звёздные войны: Эпизод 6 - Возвращение Джедая (1983)

12+ 25/05/1983 (US) • приключения, боевик, фантастика • 2h 14m

The Empire Falls...

Обзор

В заключительном шестом эпизоде «Звездных войн» Дарт Вейдер создает вторую «Звезду Смерти». Он объединяет все силы зла, чтобы с помощью этого смертоносного оружия нанести последний сокрушительный удар по повстанцам и их союзникам. Люк Скайуокер вместе с принцессой Лейей и верными дроидами R2D2 и C-3PO отправляется спасать своего друга Хана Соло, который попал в плен к отвратительному Джаббе Хатту — могущественному повелителю преступников.

George Lucas	Richard Marquand	Lawrence Kasdan
Screenplay, Story	Director	Screenplay