Movie app

“AleVoastreFS”

Aldica Victor-Mihai

Calc III 4LF791A

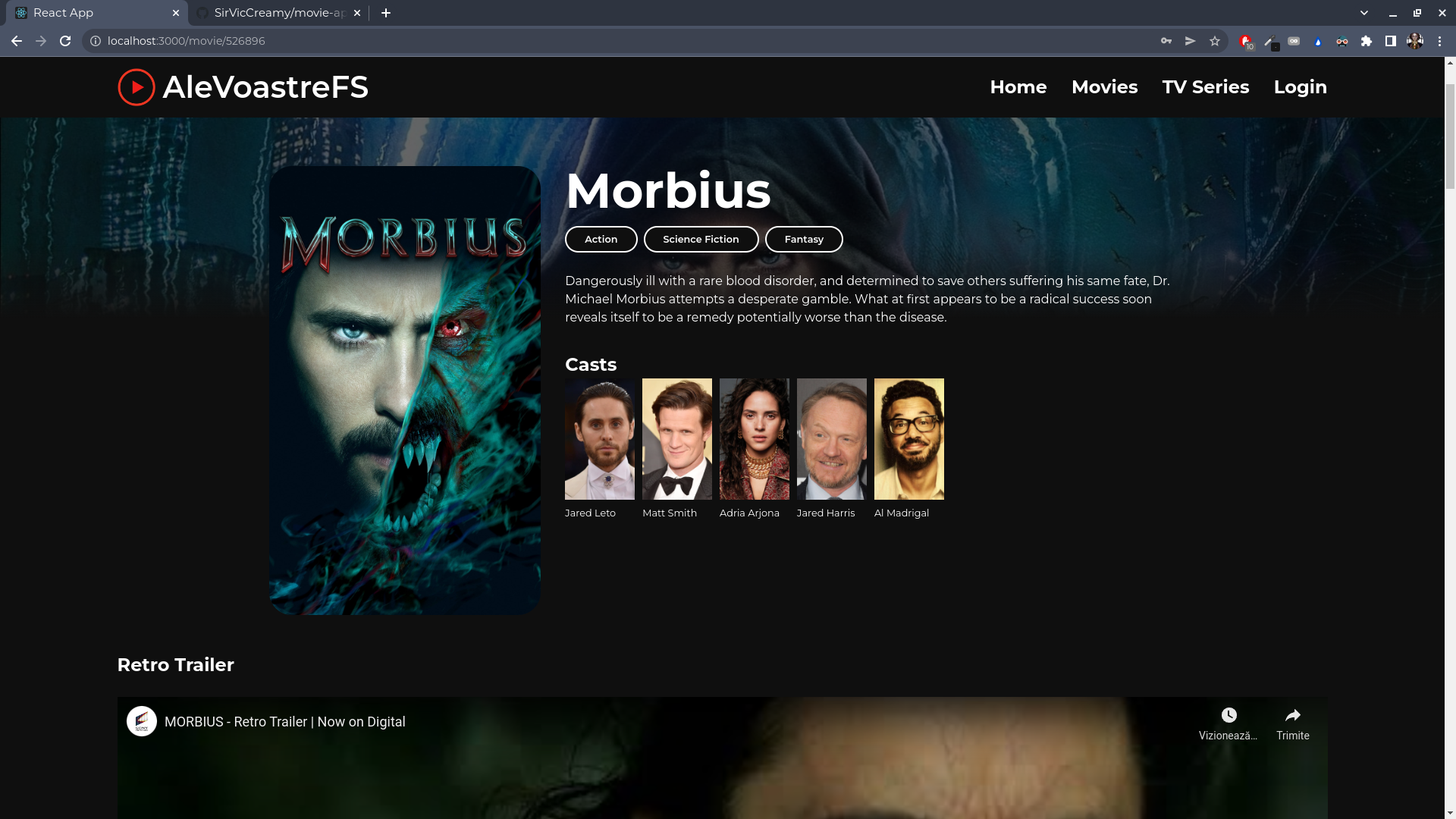
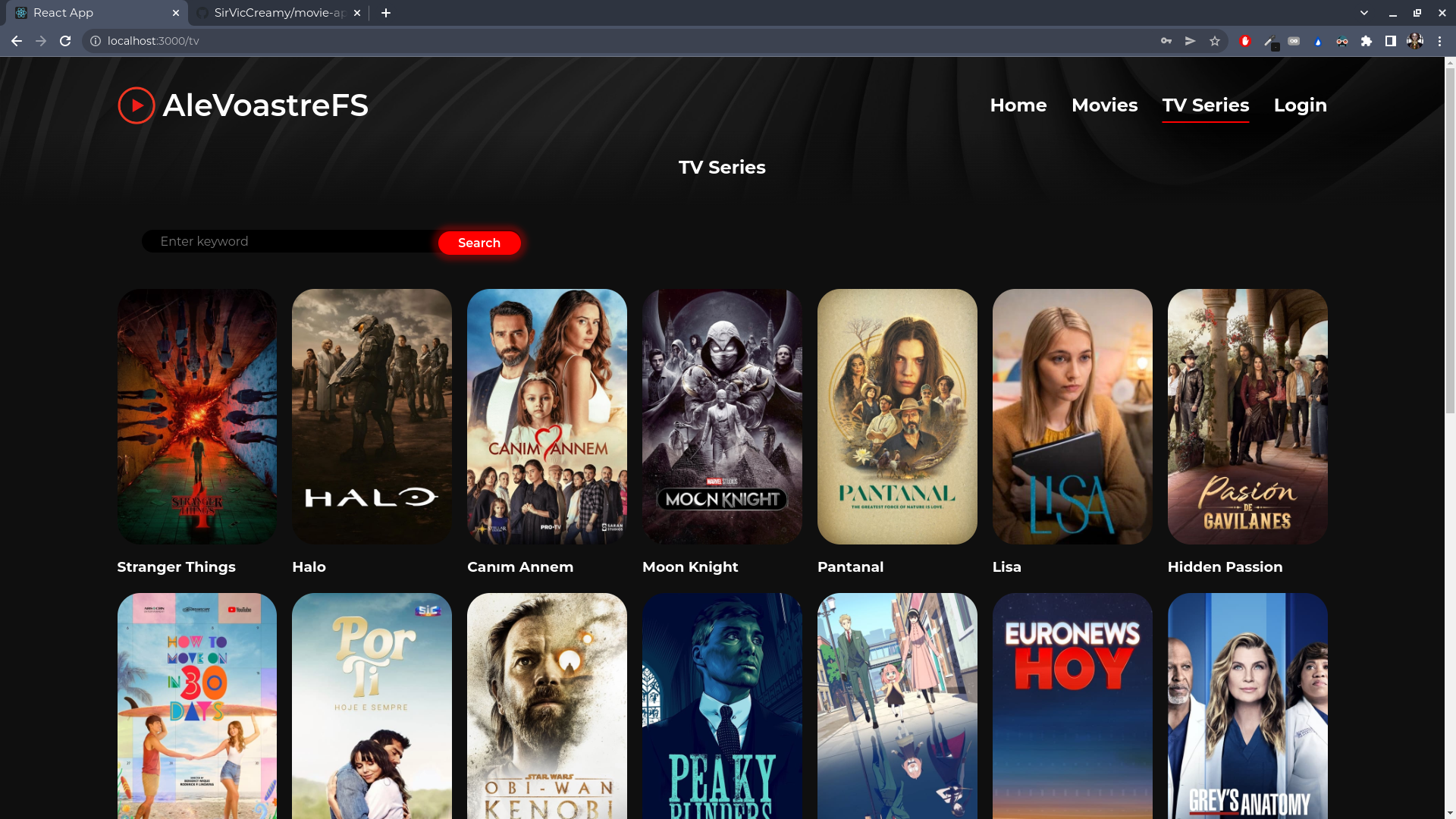
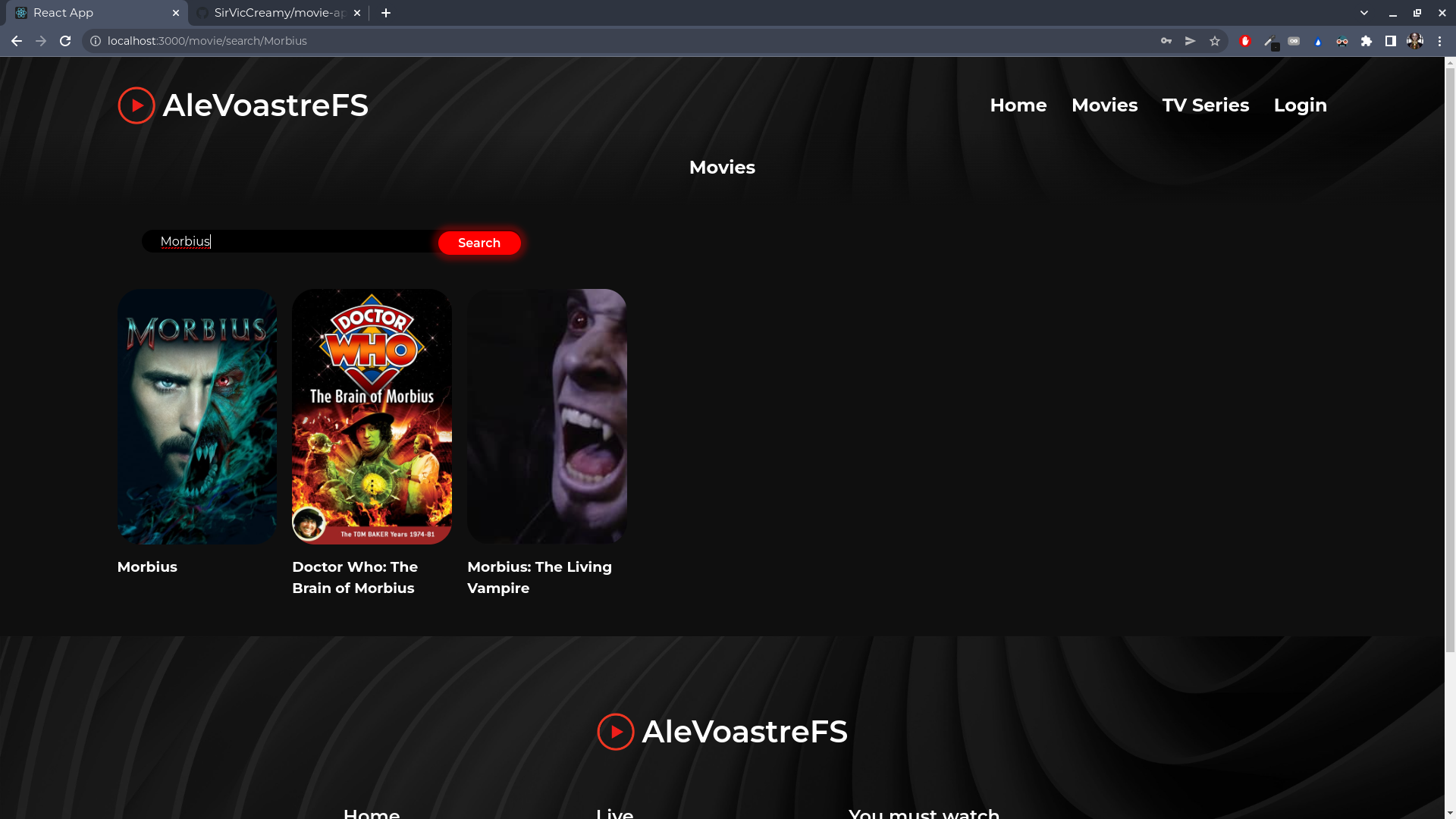
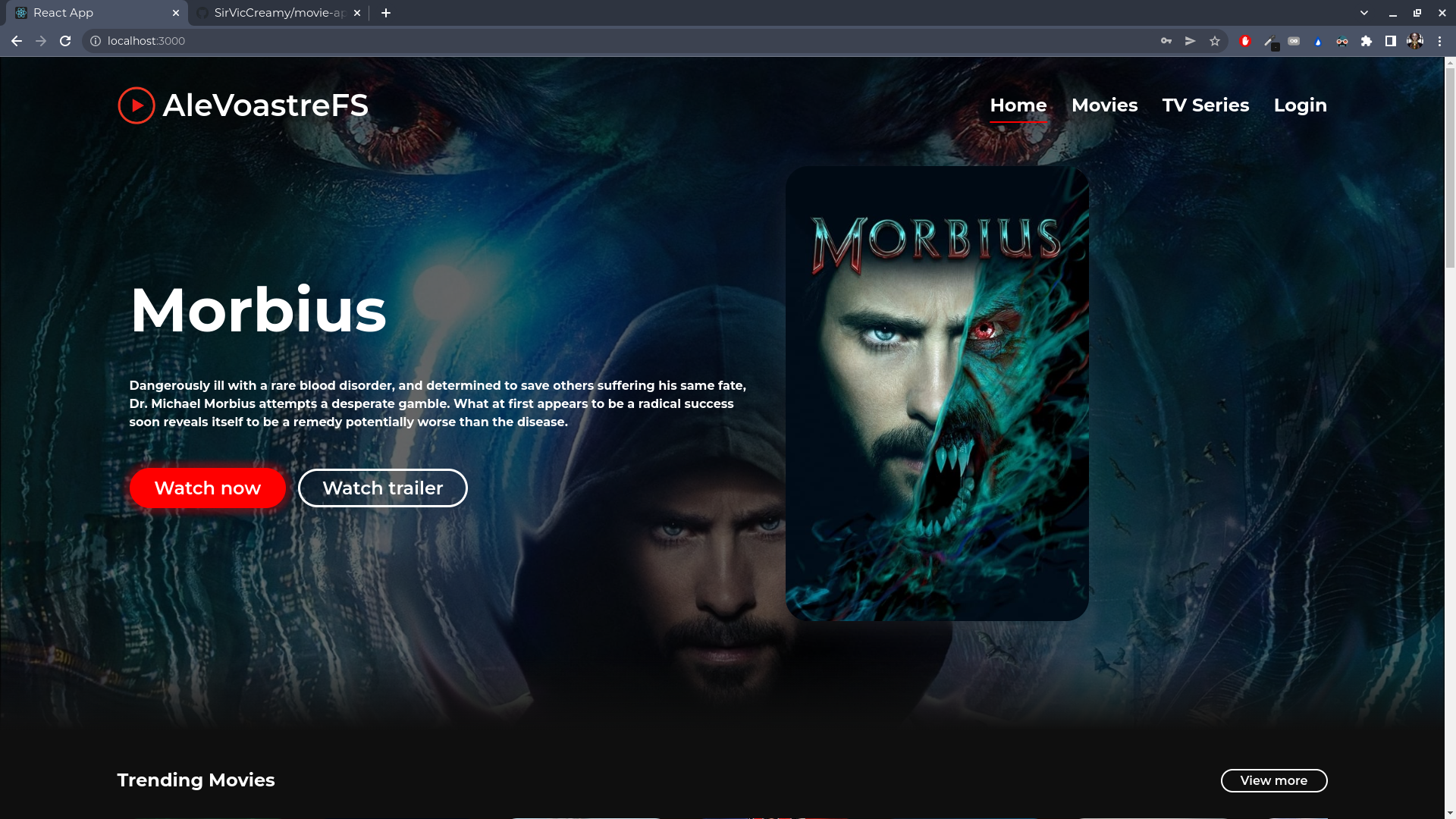
Coordonator: Dinu Ionut Vladut

Descriere:

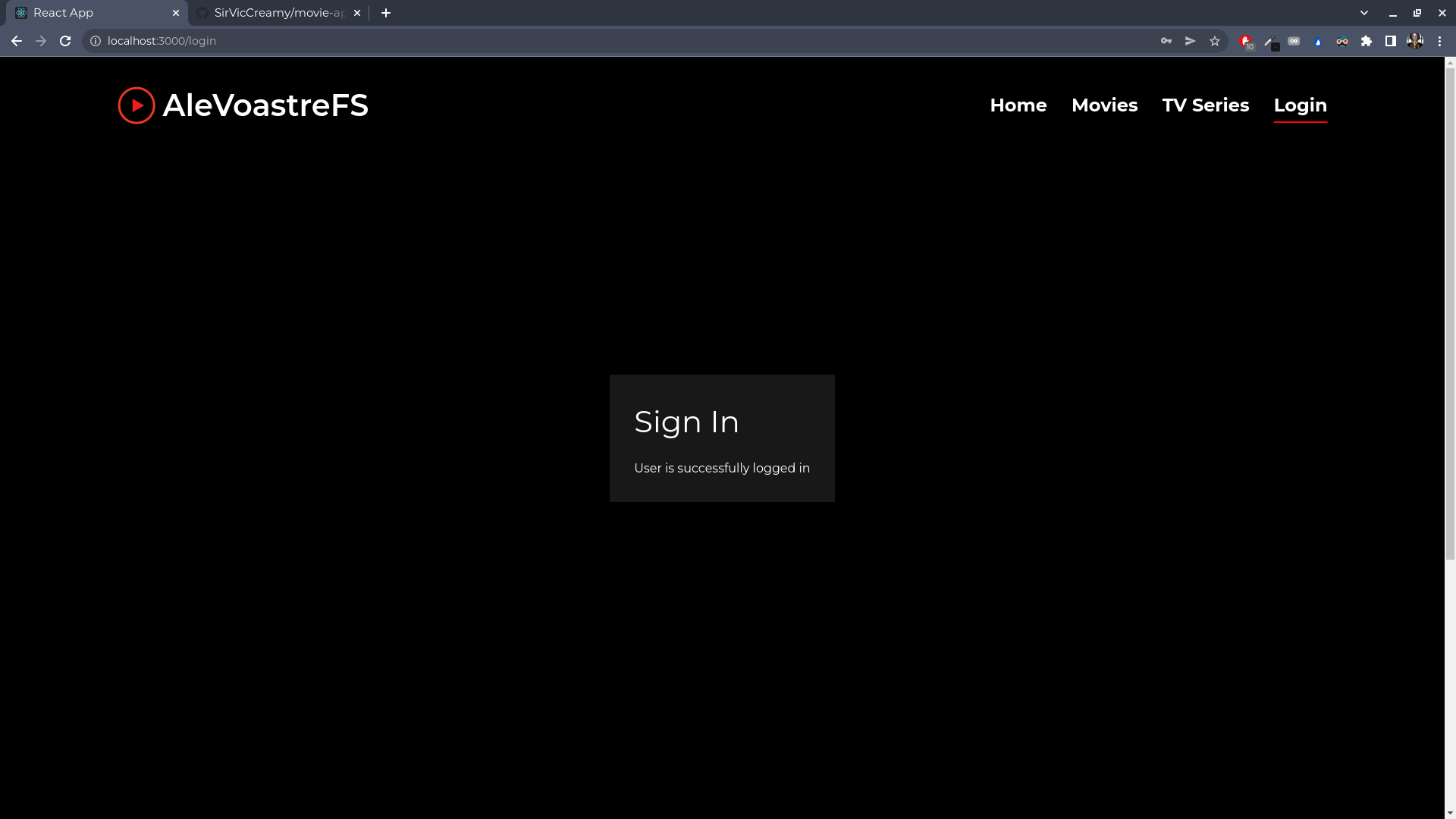
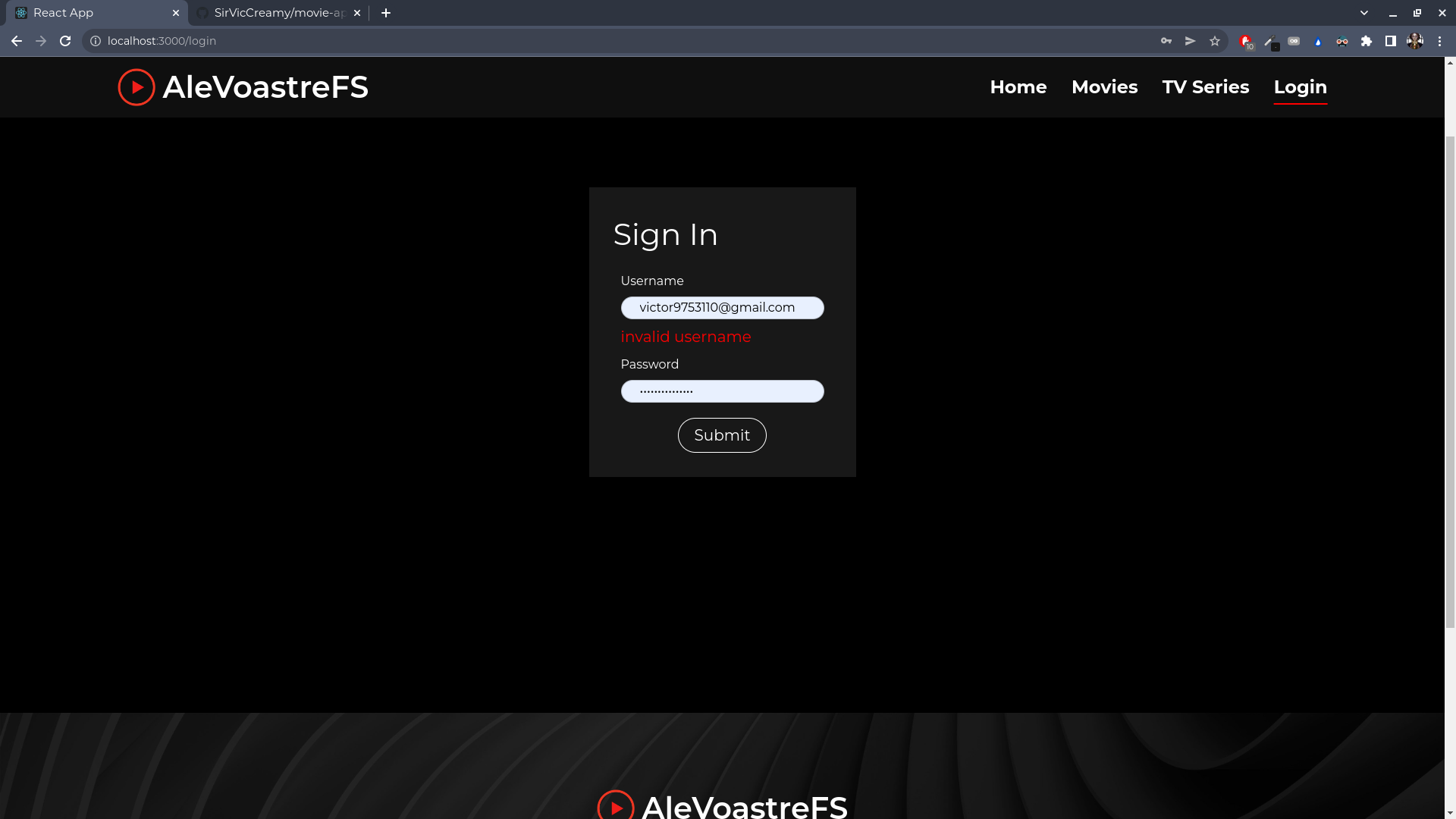
Clona a unei platforme de filme si seriale TV asemanatoare cu Netflix-ul folosind ReactJS.

Tasks:

* Utilizare git: <https://github.com/SirVicCreamy/movie-app>
* Interfata grafica:



* Logging:



* SOLID: SRP

Headerul doar gestioneaza fluxul de pagini din el

import React, { useRef, useEffect } from 'react';

import { Link, useLocation } from 'react-router-dom';

import './header.scss';

import logo from '../../assets/tmovie.png';

const headerNav = [

{

display: 'Home',

path: '/'

},

{

display: 'Movies',

path: '/movie'

},

{

display: 'TV Series',

path: '/tv'

},

{

display: 'Login',

path: '/login'

}

];

const Header = () => {

const { pathname } = useLocation();

const headerRef = useRef(null);

const active = headerNav.findIndex(e => e.path === pathname);

useEffect(() => {

const shrinkHeader = () => {

if (document.body.scrollTop > 100 || document.documentElement.scrollTop > 100) {

headerRef.current.classList.add('shrink');

} else {

headerRef.current.classList.remove('shrink');

}

}

window.addEventListener('scroll', shrinkHeader);

return () => {

window.removeEventListener('scroll', shrinkHeader);

};

}, []);

return (

<div ref={headerRef} className="header">

<div className="header\_\_wrap container">

<div className="logo">

<img src={logo} alt="" />

<Link to="/">AleVoastreFS</Link>

</div>

<ul className="header\_\_nav">

{

headerNav.map((e, i) => (

<li key={i} className={`${i === active ? 'active' : ''}`}>

<Link to={e.path}>

{e.display}

</Link>

</li>

))

}

</ul>

</div>

</div>

);

}

export default Header;

* ORM (folosind baza de date [https://www.themoviedb.org/)](https://www.themoviedb.org/):

import axiosClient from "./axiosClient";

export const category = {

movie: 'movie',

tv: 'tv'

}

export const movieType = {

upcoming: 'upcoming',

popular: 'popular',

top\_rated: 'top\_rated'

}

export const tvType = {

popular: 'popular',

top\_rated: 'top\_rated',

on\_the\_air: 'on\_the\_air'

}

const tmdbApi = {

getMoviesList: (type, params) => {

const url = 'movie/' + movieType[type];

return axiosClient.get(url, params);

},

getTvList: (type, params) => {

const url = 'tv/' + tvType[type];

return axiosClient.get(url, params);

},

getVideos: (categ, id) => {

const url = category[categ] + '/' + id + '/videos';

return axiosClient.get(url, {params: {}});

},

search: (categ, params) => {

const url = 'search/' + category[categ];

return axiosClient.get(url, params);

},

detail: (categ, id, params) => {

const url = category[categ] + '/' + id;

return axiosClient.get(url, params);

},

credits: (categ, id) => {

const url = category[categ] + '/' + id + '/credits';

return axiosClient.get(url, {params: {}});

},

similar: (categ, id) => {

const url = category[categ] + '/' + id + '/similar';

return axiosClient.get(url, {params: {}});

},

}

export default tmdbApi;

* MVC: ex Routing controller

import React from 'react';

import { Route, Switch } from 'react-router-dom';

import Home from '../pages/Home';

import Catalog from '../pages/Catalog';

import Detail from '../pages/detail/Detail';

import Login from '../pages/Login';

const Routes = () => {

return (

<Switch>

<Route

path='/login'

component={Login}

/>

<Route

path='/:category/search/:keyword'

component={Catalog}

/>

<Route

path='/:category/:id'

component={Detail}

/>

<Route

path='/:category'

component={Catalog}

/>

<Route

path='/'

exact

component={Home}

/>

</Switch>

);

}

export default Routes;

* Design pattern: Singleton

import React, { useState } from "react";

import "./detail/login.scss";

function Login() {

// React States

const [errorMessages, setErrorMessages] = useState({});

const [isSubmitted, setIsSubmitted] = useState(false);

// User Login info

const database = [

{

username: "user1",

password: "pass1"

},

{

username: "user2",

password: "pass2"

}

];

const errors = {

uname: "invalid username",

pass: "invalid password"

};

const handleSubmit = (event) => {

//Prevent page reload

event.preventDefault();

var { uname, pass } = document.forms[0];

// Find user login info

const userData = database.find((user) => user.username === uname.value);

// Compare user info

if (userData) {

if (userData.password !== pass.value) {

// Invalid password

setErrorMessages({ name: "pass", message: errors.pass });

} else {

setIsSubmitted(true);

}

} else {

// Username not found

setErrorMessages({ name: "uname", message: errors.uname });

}

};

// Generate JSX code for error message

const renderErrorMessage = (name) =>

name === errorMessages.name && (

<div className="error">{errorMessages.message}</div>

);

// JSX code for login form

const renderForm = (

<div className="form">

<form onSubmit={handleSubmit}>

<div className="input-container">

<label>Username </label>

<input type="text" name="uname" required />

{renderErrorMessage("uname")}

</div>

<div className="input-container">

<label>Password </label>

<input type="password" name="pass" required />

{renderErrorMessage("pass")}

</div>

<div className="button-container">

<input type="submit" />

</div>

</form>

</div>

);

return (

<div className="login">

<div className="login-form">

<div className="title">Sign In</div>

{isSubmitted ? <div>User is successfully logged in</div> : renderForm}

</div>

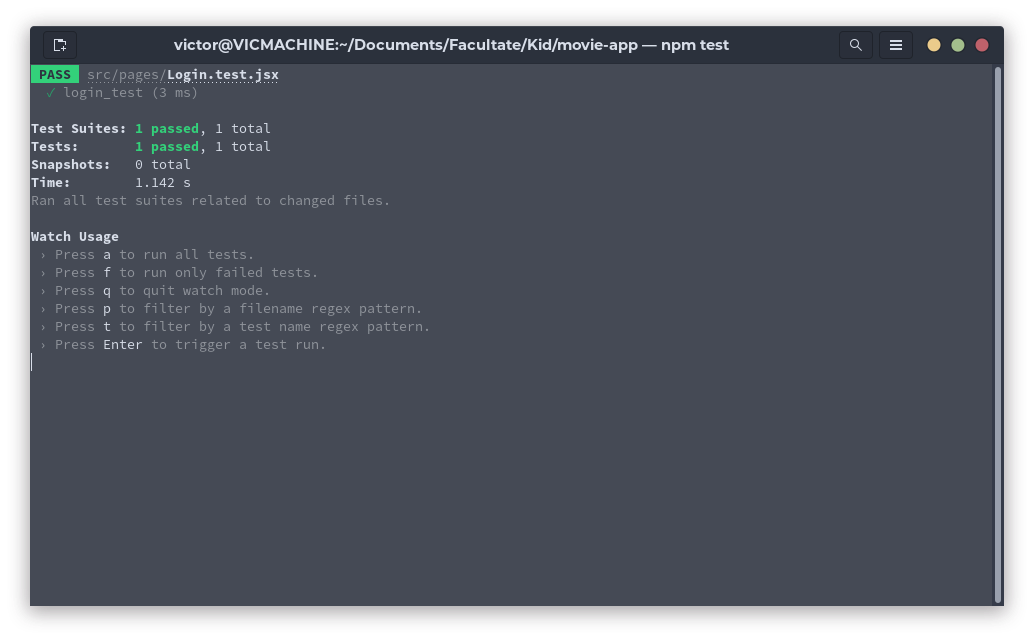
</div>

);

}

export default Login;

* Unit testing:



* UML:

<https://github.com/SirVicCreamy/movie-app/blob/master/url.svg>