**Documents – JFSD PROJECTS**

**Course – 1**

**Streamify**

**Source Code**

* **By Santhanalakshmi**

**Movies.js (Path : src/Components/Movies.js )**

import axios from "axios";

import { useContext, useEffect, useState } from "react";

import { Row, Col, CardGroup, Card, Container, CardFooter } from "react-bootstrap";

import "../Css/Main.css"

import ScaleLoader from "react-spinners/ScaleLoader";

import api from "../Api";

import { NavDropdown } from "react-bootstrap";

import { useSelector } from "react-redux";

function Movies() {

let [Loading, setLoading] = useState("false");

let [movielist, SetMovieList] = useState([{}]);

let [Error, setError] = useState("false");

let [result, setResult] = useState([{}]);

let [text, setText] = useState("");

let data = useSelector(gv => gv.genres);

let handleSearch = (event) => {

axios.get(`${api.api\_genre\_url}` + text)

.then(Response => { setResult(Response.data.results) })

.catch(e => console.log(e))

}

useEffect(() => {

axios.get(`${api.api\_search\_url}`)

.then(Response => { SetMovieList(Response.data.results) })

.catch(e => alert("Error while Fetching data"))

setLoading(true);

setTimeout(() => {

setLoading(false);

}, 2000);

}, [])

return (

<div>

<header>

<NavDropdown title="Categories" id="nav-dropdown" className="nav">

{

data.map(p =>

<NavDropdown.Item href={'/' + p.id + '/' + p.name} value={p.id}>{p.name}</NavDropdown.Item>

)

}

</NavDropdown>

<form id="form" onSubmit={handleSearch()}>

<input type="text" id="search" placeholder="Search Movies" className="search"

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

</form>

</header>

{

Loading ? <ScaleLoader

color={'#a83283'}

loading={Loading}

size={150}

aria-label="Loading Spinner"

data-testid="loader"

/> : (text !== '' && result == '')

? <span className="text-danger"> please provide valid search </span> :

text === '' ?

<Container>

<Row className="RowDesign">

{movielist.map(

searchValue => <Col md="4" className="equal-height-col" key={searchValue.id}>

<div className="ToolBox">

<CardGroup>

<Card >

<Card.Img variant="top" src={`https://image.tmdb.org/t/p/w500${searchValue.poster\_path}`} alt="Movies" />

<Card.Body>

<Card.Title >{searchValue.title}</Card.Title>

<CardFooter

className={searchValue.vote\_average >= 8 ?

"text-success" :

searchValue.vote\_average >= 5 ?

"text-info" : "text-danger"}

> Rating : {searchValue.vote\_average}

</CardFooter>

</Card.Body>

</Card>

</CardGroup>

</div>

</Col>

)}

</Row>

</Container> :

<Container>

<Row className="RowDesign">

{result.map(

searchValue => <Col md="4" className="equal-height-col" key={searchValue.id}>

<div className="ToolBox">

<CardGroup>

<Card >

<Card.Img variant="top" src={`https://image.tmdb.org/t/p/w500${searchValue.poster\_path}`} alt="Movies" />

<Card.Body>

<Card.Title >{searchValue.title}</Card.Title>

<CardFooter

className={searchValue.vote\_average >= 8 ?

"text-success" :

searchValue.vote\_average >= 5 ?

"text-info" : "text-danger"}

> Rating : {searchValue.vote\_average}

</CardFooter>

</Card.Body>

</Card>

</CardGroup>

</div>

</Col>

)}

</Row>

</Container>

}

</div>

)

}

export default Movies;

**Main.js (Path : src/Components/Main.js )**

import { useState } from "react"

import MyContext from "./Context"

import Header from "./Header"

import Movies from "./Movies";

import Genre from "./Genre";

import Footer from "./Footer";

function Main() {

return (

<div>

<Movies />

<Footer/>

</div>

)

}

export default Main

**Footer.js (Path : src/Components/Footer.js )**

function Footer() {

return (

<p> &copy; JFSD Course1 project by <b> Santhanalakshmi </b></p>

)

}

export default Footer

**Header.js (Path : src/Components/Header.js )**

import { useContext } from "react";

import "../Css/style.css"

import MyContext from "./Context";

function Header() {

const { text, setText } = useContext(MyContext);

return (

<div>

<header>

<form id="form">

<input type="text" id="search" placeholder="Search Movies" className="search"

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

</form>

</header>

</div>

)

}

export default Header;

**Genrelist.js (Path : src/Components/Genrelist.js )**

import axios from "axios";

import { useContext, useEffect, useState } from "react";

import { Row, Col, CardGroup, Card, Container, CardFooter } from "react-bootstrap";

import "../Css/Main.css"

import ScaleLoader from "react-spinners/ScaleLoader";

import MyContext from "./Context";

import { useLocation } from "react-router";

import api from "../Api";

function Genrelist() {

let [Loading, setLoading] = useState("false");

let [movielist, SetMovieList] = useState([{}]);

let [error, setError] = useState("false")

let location = useLocation();

let GenreName = location.pathname.split('/')[2];

let data = useContext(MyContext);

useEffect(() => {

axios.get(`${api.api\_genre\_url}` + GenreName)

.then(Response => { SetMovieList(Response.data.results) })

.catch(e => console.log(e))

setLoading(true);

setTimeout(() => {

setLoading(false);

}, 2000);

}, [])

return (

<div>

{Loading ? <ScaleLoader

color={'#a83283'}

loading={Loading}

size={150}

aria-label="Loading Spinner"

data-testid="loader"

/> :

<Container>

<Row className="RowDesign">

{movielist.filter((searchValue => {

return data.text.toLowerCase() === '' ? searchValue :

(searchValue.title.toLowerCase().includes(data.text))

})).map(

searchValue => <Col md="4" className="equal-height-col" key={searchValue.id}>

<div className="ToolBox">

<CardGroup>

<Card >

<Card.Img variant="top" src={`https://image.tmdb.org/t/p/w500${searchValue.poster\_path}`} alt="Movies" />

<Card.Body>

<Card.Title >{searchValue.title}</Card.Title>

<CardFooter

className={searchValue.vote\_average >= 8 ?

"text-success" :

searchValue.vote\_average >= 5 ?

"text-info" : "text-danger"}

> Rating : {searchValue.vote\_average}

</CardFooter>

</Card.Body>

</Card>

</CardGroup>

</div>

</Col>

)}

</Row>

</Container>

}

</div>

)

}

export default Genrelist;

**Genre.js (Path : src/Components/Genre.js )**

import Header from "./Header";

import Genrelist from "./Genrelist";

import Footer from "./Footer";

import { useState } from "react";

import MyContext from "./Context";

function Genre() {

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

return (

<div>

<MyContext.Provider value={{ text, setText }}>

<Header />

<Genrelist />

</MyContext.Provider>

<Footer />

</div>

)

}

export default Genre;

**Context.js (Path : src/Components/Context.js )**

import { createContext } from "react";

let MyContext = createContext();

export default MyContext;

**Main.css (Path : src/Css/Main.css )**

@media screen and (max-width:1450px) {

.ToolBox {

position: relative;

margin-top: 10%;

margin-left: 10%;

}

.RowDesign {

padding-top: 3%;

margin-right: 3%;

overflow: auto;

}

}

@media screen and (max-width:1500px) {

.ToolBox {

position: relative;

margin-top: 7%;

margin-left: 10%;

}

.RowDesign {

padding-top: 5%;

margin-right: 7%;

height: 100;

}

}

.Genre {

display: flex;

align-items: center;

justify-content: center;

padding: 0.5rem 1rem 1rem;

letter-spacing: 0.5px;

}

**style.css (Path : src/Css/style.css )**

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@200;400;700&display=swap');

\*{

box-sizing: border-box;

}

:root{

--primary-color:#22254b;

--secondary-color: #373b69;

}

body{

background-color: var(--primary-color);

font-family: 'Poppins', sans-serif;

margin: 0;

}

header{

padding:1rem;

display:flex;

justify-content: flex-end;

background-color: var(--secondary-color);

}

.search{

background-color: transparent;

border: 2px solid var(--primary-color);

padding:0.5rem 1rem;

border-radius: 50px;

font-size: 1rem;

color:#fff;

font-family: inherit;

}

.nav{

background-color: transparent;

border: 1px solid var(--secondary-color);

padding:0.3rem 1rem;

border-radius: 40px;

font-size: 1rem;

color:#691111;

font-family: inherit;

align-self: flex-start;

}

.search:focus{

outline:0;

background-color: var(--primary-color);

}

.search::placeholder{

color: #c5a273;

}

#nav-dropdown{

color: #c5a273;

}

.nav:focus{

outline:0;

background-color: var(--secondary-color);

}

main{

display:flex;

flex-wrap: wrap;

justify-content: center;

}

.movie {

width: 300px;

margin: 1rem;

border-radius: 3px;

box-shadow: 0.2px 4px 5px rgba(0,0,0,0.1);

background-color: var(--secondary-color);

position:relative;

overflow: hidden;

}

.movie img{

width:100%;

}

.movie-info{

color:#eee;

display: flex;

align-items: center;

justify-content: space-between;

padding: 0.5rem 1rem 1rem;

letter-spacing: 0.5px;

}

.movie-info h3{

margin-top: 0;

}

.movie-info span{

background-color: var(--primary-color);

padding: 0.25rem 0.5rem;

border-radius: 3px;

font-weight: bold;

}

.movie-info span.green{

color:lightgreen;

}

.movie-info span.orange{

color:orange;

}

.movie-info span.red{

color:red;

}

.overview{

position:absolute;

left:0;

right:0;

bottom:0;

background-color: #fff;

padding: 1rem;

max-height: 100%;

transform:translateY(101%);

transition:transform 0.3s ease-in;

}

.movie:hover .overview{

transform:translateY(0)

}

#tags{

width:80%;

display:flex;

flex-wrap: wrap;

justify-content: center;

align-items: center;

margin: 10px auto;

}

.tag{

color:white;

padding:10px 20px;

background-color: orange;

border-radius: 50px;

margin:5px;

display:inline-block;

cursor: pointer;

}

.tag.highlight{

background-color: red;

}

.no-results{

color:white;

}

.pagination{

display:flex;

margin:10px 30px;

align-items: center;

justify-content: center;

color:white;

}

.page{

padding:20px;

cursor:pointer;

}

.page.disabled{

cursor:not-allowed;

color:grey;

}

.current{

padding: 10px 20px;

border-radius: 50%;

border: 5px solid orange;

font-size: 20px;

font-weight: 600;

}

.know-more{

background-color: orange;

color:white;

font-size: 16px;

font-weight: bold;

border:0;

border-radius: 50px;

padding: 10px 20px;

margin-top: 5px;

}

/\* The Overlay (background) \*/

.overlay {

/\* Height & width depends on how you want to reveal the overlay (see JS below) \*/

height: 100%;

width: 0;

position: fixed; /\* Stay in place \*/

z-index: 1; /\* Sit on top \*/

left: 0;

top: 0;

background-color: rgb(0,0,0); /\* Black fallback color \*/

background-color: rgba(0,0,0, 0.9); /\* Black w/opacity \*/

overflow-x: hidden; /\* Disable horizontal scroll \*/

transition: 0.5s; /\* 0.5 second transition effect to slide in or slide down the overlay (height or width, depending on reveal) \*/

}

/\* Position the content inside the overlay \*/

.overlay-content {

position: relative;

top: 10%; /\* 25% from the top \*/

width: 100%; /\* 100% width \*/

text-align: center; /\* Centered text/links \*/

margin-top: 30px; /\* 30px top margin to avoid conflict with the close button on smaller screens \*/

}

/\* The navigation links inside the overlay \*/

.overlay a {

padding: 8px;

text-decoration: none;

font-size: 36px;

color: #818181;

display: block; /\* Display block instead of inline \*/

transition: 0.3s; /\* Transition effects on hover (color) \*/

}

/\* When you mouse over the navigation links, change their color \*/

.overlay a:hover, .overlay a:focus {

color: #f1f1f1;

}

/\* Position the close button (top right corner) \*/

.overlay .closebtn {

position: absolute;

top: 20px;

right: 45px;

font-size: 60px;

}

/\* When the height of the screen is less than 450 pixels, change the font-size of the links and position the close button again, so they don't overlap \*/

@media screen and (max-height: 450px) {

.overlay a {font-size: 20px}

.overlay .closebtn {

font-size: 40px;

top: 15px;

right: 35px;

}

}

.embed.hide{

display: none;

}

.embed.show{

display: inline-block;

}

.arrow{

position: absolute;

font-size: 40px;

}

.arrow.left-arrow{

top:50%;

left:5%;

transform: translateY(-50%);

}

.arrow.right-arrow{

top:50%;

transform: translateY(-50%);

right:5%;

}

.dots{

margin-top: 30px;

}

.dots .dot {

padding: 5px 15px;

border-radius: 50%;

border: 5px solid var(--secondary-color);

color: white;

font-size: 20px;

font-weight: 600;

margin: 5px;

}

.dots .dot.active{

border-color: orange;

}

**Reducer.js (Path : src/Redux/Reducer.js )**

let initialState = {

genres:[{"id":28,"name":"Action"},{"id":12,"name":"Adventure"},{"id":16,"name":"Animation"},{"id":35,"name":"Comedy"},{"id":80,"name":"Crime"},{"id":99,"name":"Documentary"},{"id":18,"name":"Drama"},{"id":10751,"name":"Family"},{"id":14,"name":"Fantasy"},{"id":36,"name":"History"},{"id":27,"name":"Horror"},{"id":10402,"name":"Music"},{"id":9648,"name":"Mystery"},{"id":10749,"name":"Romance"},{"id":878,"name":"Science Fiction"},{"id":10770,"name":"TV Movie"},{"id":53,"name":"Thriller"},{"id":10752,"name":"War"},{"id":37,"name":"Western"}]

}

function Reducer(state=initialState,action)

{

return state ;

}

export default Reducer;

**Api.js (Path : src/Api.js )**

let API\_KEY = '&api\_key=4c9856363822bdb39f9cd73df304edfa';

let SEARCH\_URL = "/discover/movie?include\_adult=false&include\_video=false&language=en-US&page=1&sort\_by=popularity.desc";

let BASE\_URL = 'https://api.themoviedb.org/3';

let SEARCH\_QUERY = '/search/movie?include\_adult=false&language=en-US&page=1';

const api = {

api\_search\_url : BASE\_URL + SEARCH\_URL + API\_KEY,

api\_genre\_url : BASE\_URL + SEARCH\_QUERY + API\_KEY + '&query=',

}export default api

**App.js (Path : src/App.js )**

import logo from './logo.svg';

import './App.css';

import Main from './Components/Main';

import { useSelector } from 'react-redux';

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

import Genre from './Components/Genre';

function App() {

let data = useSelector(gv => gv.genres);

return (

<div className="App">

<BrowserRouter>

<Routes>

<Route exact path="/" element={<Main />}>

</Route>

</Routes>

{

data.map(p => <Routes key={p.id}>

<Route path={'/' + p.id + '/' + p.name} element={<Genre />} />

</Routes>

)

}

</BrowserRouter>

</div>

);

}

export default App;

**Index.js (Path : src/Index.js )**

import React from 'react';

import ReactDOM from 'react-dom/client';

import './index.css';

import App from './App';

import reportWebVitals from './reportWebVitals';

import "bootstrap/dist/css/bootstrap.min.css";

import "bootstrap/dist/js/bootstrap.min.js";

import { legacy\_createStore as createStore } from 'redux';

import Reducer from './Redux/Reducer';

import { Provider } from 'react-redux';

const storeref = createStore(Reducer);

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(

<React.StrictMode>

<Provider store={storeref}>

<App />

</Provider>

</React.StrictMode>

);

// 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();

**Package.json**

{

"name": "react\_streamify",

"version": "0.1.0",

"private": true,

"dependencies": {

"@testing-library/jest-dom": "^5.17.0",

"@testing-library/react": "^13.4.0",

"@testing-library/user-event": "^13.5.0",

"axios": "^1.7.2",

"bootstrap": "^5.3.3",

"bootstrap-icons": "^1.11.3",

"formik": "^2.4.6",

"react": "^18.3.1",

"react-bootstrap": "^2.10.2",

"react-dom": "^18.3.1",

"react-redux": "^9.1.2",

"react-router": "^6.23.1",

"react-router-dom": "^6.23.1",

"react-scripts": "5.0.1",

"react-spinners": "^0.13.8",

"redux": "^5.0.1",

"web-vitals": "^2.1.4"

},

"scripts": {

"start": "react-scripts start",

"build": "react-scripts build",

"test": "react-scripts test",

"eject": "react-scripts eject"

},

"eslintConfig": {

"extends": [

"react-app",

"react-app/jest"

]

},

"browserslist": {

"production": [

">0.2%",

"not dead",

"not op\_mini all"

],

"development": [

"last 1 chrome version",

"last 1 firefox version",

"last 1 safari version"

]

}

}