TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*A picture containing shape

Description automatically generatedLogo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

-----------------

Môn học: Lập trình sử dụng API

Đề tài: **Trang web review phim**

Giảng viên HD: **Đinh Công Tùng**

Sinh viên thực hiện: Hà Hoàng Oanh – 201200267

Lớp CNTT2 - K61

Năm học: 2022-2023

**MỤC LỤC**

[**LỜI NÓI ĐẦU** 3](#_Toc132727748)

[**I. TỔNG QUAN VỀ ĐỀ TÀI** 4](#_Toc132727749)

[**II. THIẾT KẾ HỆ THỐNG** 5](#_Toc132727750)

[1. Giao diện đăng nhập 5](#_Toc132727751)

[2. Giao diện đăng kí 5](#_Toc132727752)

[3. Giao diện trang chủ 6](#_Toc132727753)

[4. Giao diện TV Shows 6](#_Toc132727754)

[5. Giao diện Movies 7](#_Toc132727755)

[**III. CÁC BƯỚC TRIỂN KHAI DỰ ÁN VÀ CODE** 8](#_Toc132727756)

[**KẾT LUẬN** 38](#_Toc132727757)

[**HƯỚNG PHÁT TRIỂN** 38](#_Toc132727758)

[**TÀI LIỆU THAM KHẢO** 39](#_Toc132727759)

# **LỜI NÓI ĐẦU**

Trong thời đại của công nghệ số, trang web đóng vai trò vô cùng quan trọng trong việc quảng bá sản phẩm và dịch vụ của một doanh nghiệp. Đặc biệt, với lĩnh vực giải trí như xem phim, trang web review phim trở thành một nơi để người dùng có thể tìm kiếm thông tin về các bộ phim trước khi quyết định xem. Việc tạo ra một trang web review phim chất lượng với các tính năng tiện ích và thông tin phong phú sẽ giúp thu hút người dùng và tạo ra sự khác biệt so với các trang web cùng lĩnh vực khác.

Với đề tài lập trình sử dụng API để tạo trang web review phim, em đã đặt mục tiêu tạo ra một trang web cơ bản, đáp ứng nhu cầu của người dùng trong việc xem thông tin về phim. Trong báo cáo này, em sẽ trình bày quá trình thiết kế và triển khai trang web review phim, cùng với các kỹ thuật và công nghệ được sử dụng.

Em xin chân thành cảm ơn sự giúp đỡ tận tình của thầy **Đinh Công Tùng**. Rất mong nhận được sự đóng góp ý kiến và đánh giá từ thầy cũng như các bạn để bài tập lớn của em được hoàn thiện tốt nhất.

**Trân trọng cảm ơn!**

# **I. TỔNG QUAN VỀ ĐỀ TÀI**

Trang web được thiết kế và xây dựng nhằm phục vụ ứng nhu cầu của người dùng trong việc xem thông tin về phim.

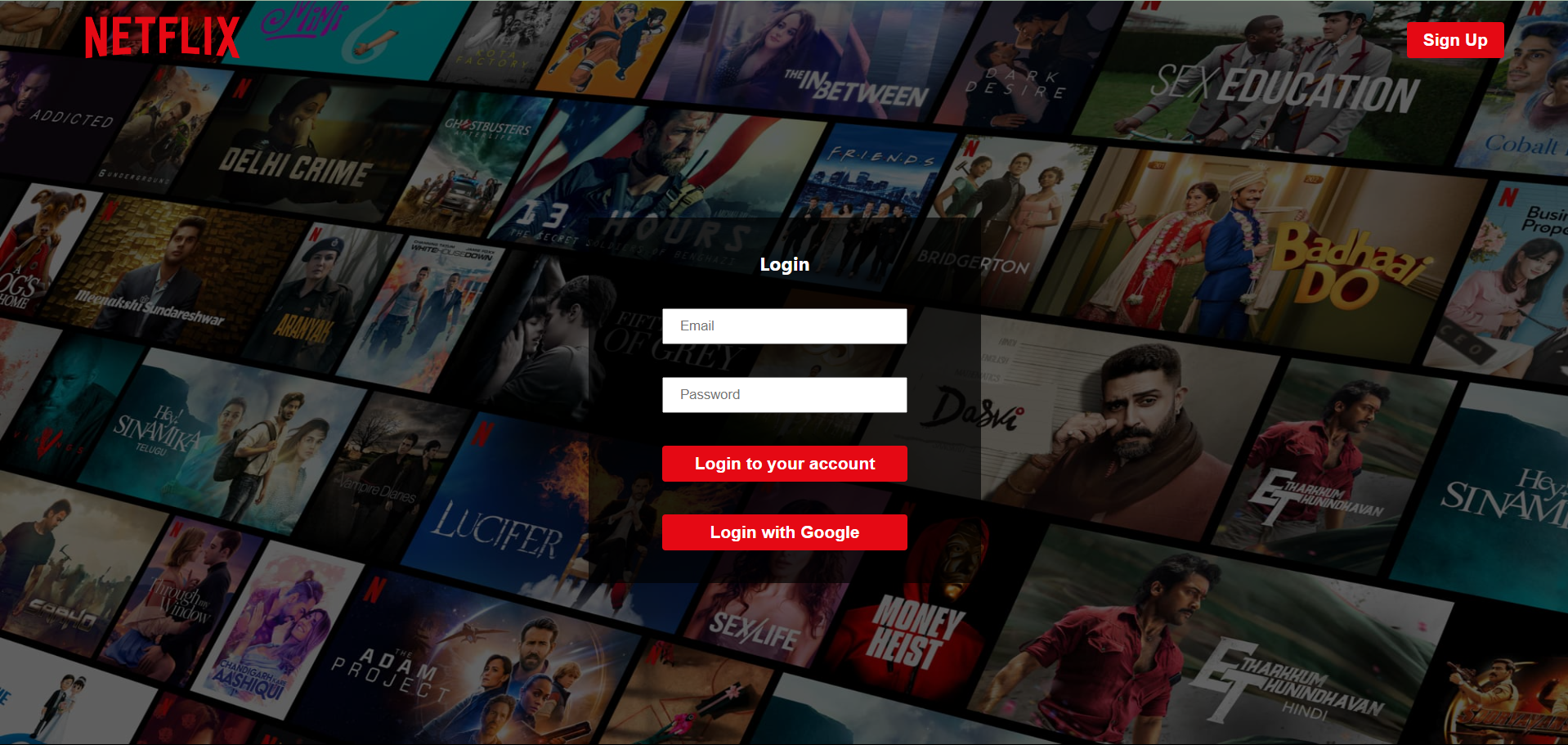
Các chức năng cơ bản hiện có gồm:

* Đăng kí tài khoản
* Đăng nhập bằng tài khoản đăng kí hoặc đăng nhập bằng tài khoản google
* Lọc Movies theo thể loại
* Lọc TV Shows theo thể loại

Trang web sử dụng một số công nghệ chính: React, Redux, Nodejs.

# **II. THIẾT KẾ HỆ THỐNG**

## 1. Giao diện đăng nhập



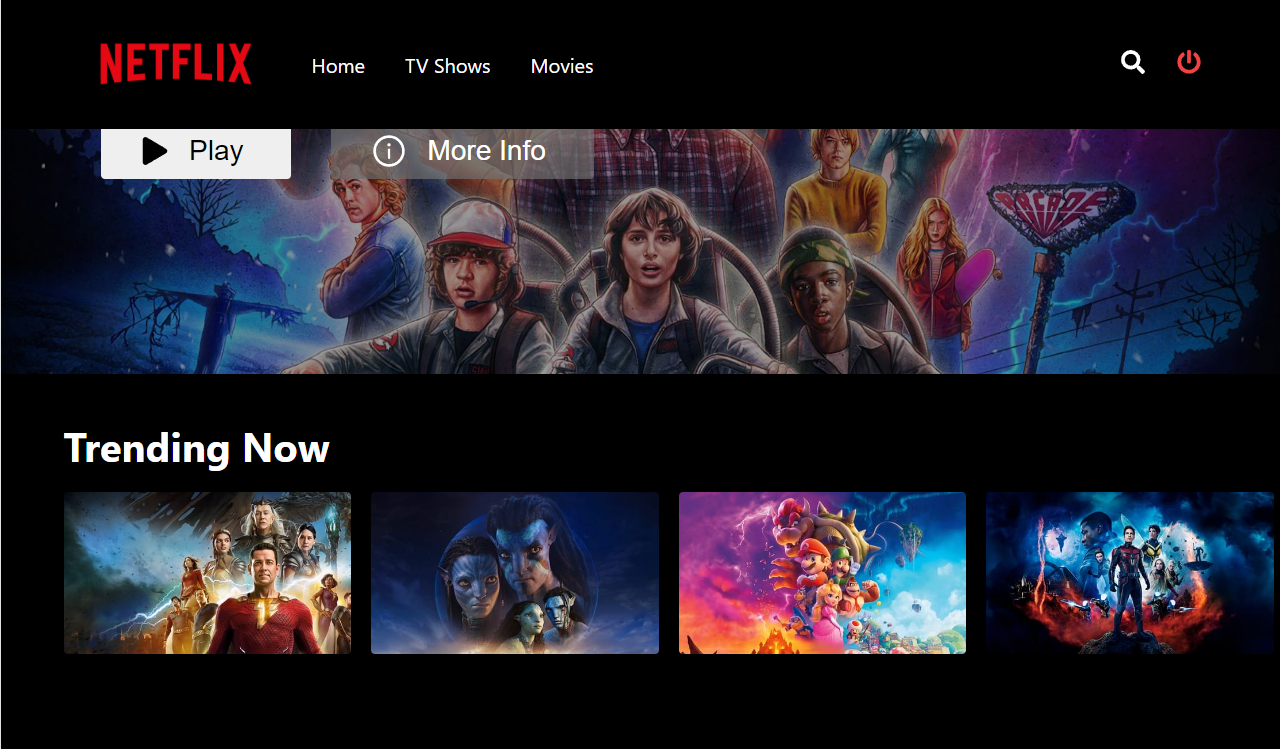
Login to your account nếu đã có tài khoản của trang web hoặc có thể đăng nhập với tài khoản google. Sau khi đăng nhập sẽ được chuyển đến giao diện trang chủ.

## 2. Giao diện đăng kí



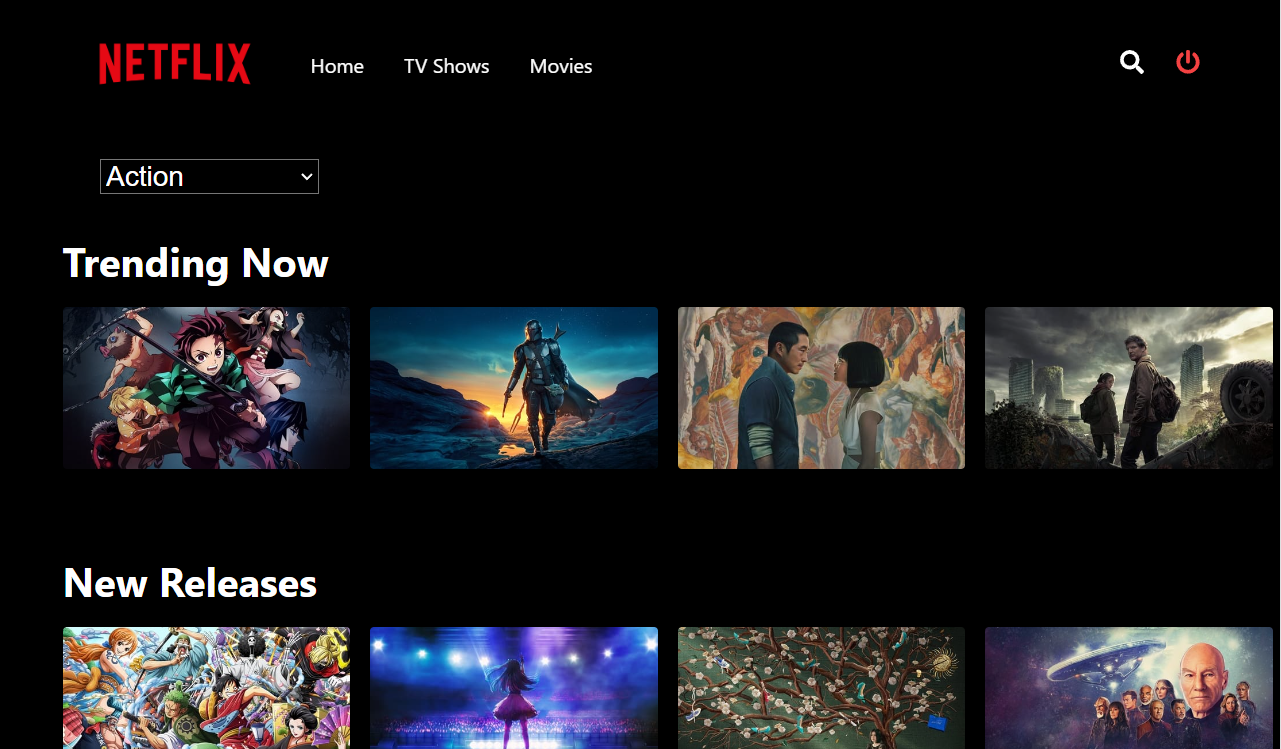
Nhập địa chỉ email chọn Get started sẽ hiện ô nhập password, nhập password sau đó bấm chọn nút Login vừa mới hiện ra để chuyển đến trang chủ.

## 3. Giao diện trang chủ



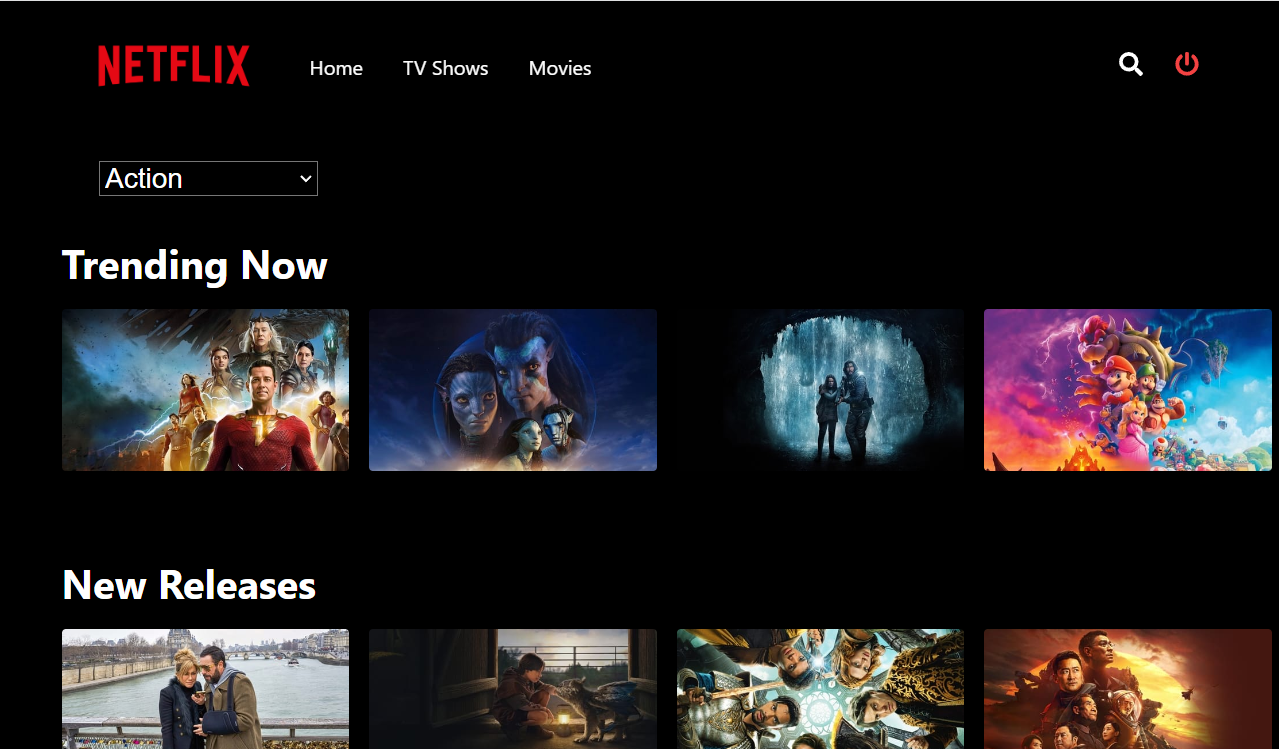
Khi di chuột vào từng phim sẽ hiện tên phim và thể loại.

## 4. Giao diện TV Shows



Chọn thể loại trong menu sẽ hiển thị các phim có thể loại tương ứng.

## 5. Giao diện Movies



Chọn thể loại trong menu sẽ hiển thị các phim có thể loại tương ứng.

# **III. CÁC BƯỚC TRIỂN KHAI DỰ ÁN VÀ CODE**

- Cài đặt Nodejs

- Tạo thư mục lưu dự án và add vào Visual studio. Tạo ứng dụng React có tên netflix-ui bằng cách sử dụng lệnh sau trong terminal: npx create-react-app@latest netflix-ui

- Di chuyển đến ứng dụng bằng lệnh: cd F:\LTAPI\netflixClone\netflix-clone-react-node\netflix-ui

- Cài đặt các thư viện, tài nguyên cho dự án với: npm install axios firebase styled-components react-icons react-redux @reactjs/toolkit react-router-dom

- Cài đặt các gói phầm mềm Nodejs bằng: npm install

- Xóa bỏ một số file không dùng đến trong ứng dụng như: App.css, App.test.js, logo.svg, reportWebVital.js, setupTest.js.

- Trong thư mục src thêm các thư mục con: assets, components, pages, store, utils.

- Thư mục assets thêm các ảnh và video dạng mp4.

- Thư mục compponents thêm các file sau: BackgroundImage.jsx, Card.jsx, CardSlider.jsx, Header.jsx, Navbar.jsx, NotAvailable.jsx, SelectGenre.jsx, Slider.jsx.

- Thư mục pages thêm các file: Login.jsx, Movies.jsx, Netflix.jsx, Player.jsx, Signup.jsx, TVShows.jsx.

- Thư mục store thêm file: index.js.

- Thư mục utils thêm file: constants.js, firebase-config.js.

- Đăng kí firebase và tạo dự án trên firebase. Trong trang của dự án chọn mục Web để register app. Sau đó copy firebase SDK vào file firebase-config.js và thêm 2 đoạn code sau: import { getAuth } from "firebase/auth"; export const firebaseAuth = getAuth(app);

- Chọn mục Authentication và get started, chọn Sign-in method, chọn Email/password và Google sau đó chọn enable và save.

- Đăng kí tài khoản trong The movie database API. Trong phần setting chọn API và create. Copy API key và base url vào file constants.js:

export const API\_KEY = "2f845f870043f419c536a7be6719a564";

export const TMDB\_BASE\_URL = "https://api.themoviedb.org/3";

- Đọc API Docs ở trang <https://developers.themoviedb.org/3> để có thể sử dụng các API của The movie database API

- Sử dụng lệnh: npm start để chạy dự án

- Code file index.js:

import React from "react";

import ReactDOM from "react-dom/client";

import { Provider } from "react-redux";

import "./index.css";

import App from "./App";

import { store } from "./store";

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

root.render(

<React.StrictMode>

<Provider store={store}>

<App />

</Provider>

</React.StrictMode>

);

- Code file index.css:

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",

"Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

background-color: black;

color: white;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

html,

body {

overflow-x: hidden;

}

body::-webkit-scrollbar {

display: none;

}

.flex {

display: flex;

}

.column {

flex-direction: column;

}

.j-between {

justify-content: space-between;

}

.j-center {

justify-content: center;

}

.a-center {

align-items: center;

}

- Code file App.js:

import React from "react";

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

import Login from "./pages/Login";

import MoviePage from "./pages/Movies";

import Netflix from "./pages/Netflix";

import Player from "./pages/Player";

import Signup from "./pages/Signup";

import TVShows from "./pages/TVShows";

export default function App() {

return (

<BrowserRouter>

<Routes>

<Route exact path="/login" element={<Login />} />

<Route exact path="/signup" element={<Signup />} />

<Route exact path="/player" element={<Player />} />

<Route exact path="/tv" element={<TVShows />} />

<Route exact path="/movies" element={<MoviePage />} />

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

</Routes>

</BrowserRouter>

);

}

- Code file index.js trong thư mục store:

import {

configureStore,

createAsyncThunk,

createSlice,

} from "@reduxjs/toolkit";

import axios from "axios";

import { API\_KEY, TMDB\_BASE\_URL } from "../utils/constants";

const initialState = {

movies: [],

genresLoaded: false,

genres: [],

};

export const getGenres = createAsyncThunk("netflix/genres", async () => {

const {

data: { genres },

} = await axios.get(

"https://api.themoviedb.org/3/genre/movie/list?api\_key=2f845f870043f419c536a7be6719a564"

);

return genres;

});

const createArrayFromRawData = (array, moviesArray, genres) => {

array.forEach((movie) => {

const movieGenres = [];

movie.genre\_ids.forEach((genre) => {

const name = genres.find(({ id }) => id === genre);

if (name) movieGenres.push(name.name);

});

if (movie.backdrop\_path)

moviesArray.push({

id: movie.id,

name: movie?.original\_name ? movie.original\_name : movie.original\_title,

image: movie.backdrop\_path,

genres: movieGenres.slice(0, 3),

});

});

};

const getRawData = async (api, genres, paging = false) => {

const moviesArray = [];

for (let i = 1; moviesArray.length < 60 && i < 10; i++) {

const {

data: { results },

} = await axios.get(`${api}${paging ? `&page=${i}` : ""}`);

createArrayFromRawData(results, moviesArray, genres);

}

return moviesArray;

};

export const fetchDataByGenre = createAsyncThunk(

"netflix/genre",

async ({ genre, type }, thunkAPI) => {

const {

netflix: { genres },

} = thunkAPI.getState();

return getRawData(

`https://api.themoviedb.org/3/discover/${type}?api\_key=${API\_KEY}&with\_genres=${genre}`,

genres

);

}

);

export const fetchMovies = createAsyncThunk(

"netflix/trending",

async ({ type }, thunkAPI) => {

const {

netflix: { genres },

} = thunkAPI.getState();

return getRawData(

`${TMDB\_BASE\_URL}/trending/${type}/week?api\_key=${API\_KEY}`,

genres,

true

);

}

);

const NetflixSlice = createSlice({

name: "Netflix",

initialState,

extraReducers: (builder) => {

builder.addCase(getGenres.fulfilled, (state, action) => {

state.genres = action.payload;

state.genresLoaded = true;

});

builder.addCase(fetchMovies.fulfilled, (state, action) => {

state.movies = action.payload;

});

builder.addCase(fetchDataByGenre.fulfilled, (state, action) => {

state.movies = action.payload;

});

},

});

export const store = configureStore({

reducer: {

netflix: NetflixSlice.reducer,

},

});

export const { setGenres, setMovies } = NetflixSlice.actions;

- Code file Login.jsx:

import React, { useState } from "react";

import styled from "styled-components";

import { useNavigate } from "react-router-dom";

import BackgroundImage from "../components/BackgroundImage";

import Header from "../components/Header";

import { onAuthStateChanged, signInWithEmailAndPassword } from "firebase/auth";

import { firebaseAuth } from "../utils/firebase-config";

import { GoogleAuthProvider, signInWithPopup } from "firebase/auth";

function Login() {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const navigate = useNavigate();

const handleLogin = async () => {

try {

await signInWithEmailAndPassword(firebaseAuth, email, password);

} catch (error) {

console.log(error.code);

}

};

const handleGoogleLogin = async () => {

const provider = new GoogleAuthProvider();

try {

await signInWithPopup(firebaseAuth, provider);

} catch (error) {

console.log(error.code);

}

};

onAuthStateChanged(firebaseAuth, (currentUser) => {

if (currentUser) {

navigate("/");

}

});

return (

<Container>

<BackgroundImage />

<div className="content">

<Header />

<div className="form-container flex column a-center j-center">

<div className="form flex column a-center j-center">

<div className="title">

<h3>Login</h3>

</div>

<div className="container flex column">

<input

type="text"

placeholder="Email"

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

value={email}

/>

<input

type="password"

placeholder="Password"

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

value={password}

/>

<button onClick={handleLogin}>Login to your account</button>

<button onClick={handleGoogleLogin}>Login with Google</button>

</div>

</div>

</div>

</div>

</Container>

);

}

const Container = styled.div`

position: relative;

.content {

position: absolute;

top: 0;

left: 0;

height: 100vh;

width: 100vw;

background-color: rgba(0, 0, 0, 0.5);

grid-template-rows: 15vh 85vh;

.form-container {

gap: 2rem;

height: 85vh;

.form {

padding: 2rem;

background-color: #000000b0;

width: 25vw;

gap: 2rem;

color: white;

.container {

gap: 2rem;

input {

padding: 0.5rem 1rem;

width: 15rem;

}

button {

padding: 0.5rem 1rem;

background-color: #e50914;

border: none;

cursor: pointer;

color: white;

border-radius: 0.2rem;

font-weight: bolder;

font-size: 1.05rem;

}

}

}

}

}

`;

export default Login;

- Code file Movies.jsx:

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

import styled from "styled-components";

import Navbar from "../components/Navbar";

import { onAuthStateChanged } from "firebase/auth";

import { firebaseAuth } from "../utils/firebase-config";

import { useNavigate } from "react-router-dom";

import { useSelector, useDispatch } from "react-redux";

import { fetchMovies, getGenres } from "../store";

import SelectGenre from "../components/SelectGenre";

import Slider from "../components/Slider";

import NotAvailable from "../components/NotAvailable";

function MoviePage() {

const [isScrolled, setIsScrolled] = useState(false);

const movies = useSelector((state) => state.netflix.movies);

const genres = useSelector((state) => state.netflix.genres);

const genresLoaded = useSelector((state) => state.netflix.genresLoaded);

const navigate = useNavigate();

const dispatch = useDispatch();

useEffect(() => {

dispatch(getGenres());

}, [dispatch]);

useEffect(() => {

if (genresLoaded) {

dispatch(fetchMovies({ genres, type: "movie" }));

}

}, [dispatch, genres, genresLoaded]);

const [setUser] = useState(undefined);

onAuthStateChanged(firebaseAuth, (currentUser) => {

if (currentUser) setUser(currentUser.uid);

else navigate("/login");

});

window.onscroll = () => {

setIsScrolled(window.pageYOffset === 0 ? false : true);

return () => (window.onscroll = null);

};

return (

<Container>

<div className="navbar">

<Navbar isScrolled={isScrolled} />

</div>

<div className="data">

<SelectGenre genres={genres} type="movie" />

{movies.length ? <Slider movies={movies} /> : <NotAvailable />}

</div>

</Container>

);

}

const Container = styled.div`

.data {

margin-top: 8rem;

.not-available {

text-align: center;

color: white;

margin-top: 4rem;

}

}

`;

export default MoviePage;

- Code file Netflix.jsx:

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

import styled from "styled-components";

import Navbar from "../components/Navbar";

import backgroundImage from "../assets/home.jpg";

import MovieLogo from "../assets/homeTitle.webp";

import { onAuthStateChanged } from "firebase/auth";

import { firebaseAuth } from "../utils/firebase-config";

import { useNavigate } from "react-router-dom";

import { useSelector, useDispatch } from "react-redux";

import { fetchMovies, getGenres } from "../store";

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

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

import Slider from "../components/Slider";

function Netflix() {

const [isScrolled, setIsScrolled] = useState(false);

const movies = useSelector((state) => state.netflix.movies);

const genres = useSelector((state) => state.netflix.genres);

const genresLoaded = useSelector((state) => state.netflix.genresLoaded);

const navigate = useNavigate();

const dispatch = useDispatch();

useEffect(() => {

dispatch(getGenres());

}, [dispatch]);

useEffect(() => {

if (genresLoaded) {

dispatch(fetchMovies({ genres, type: "all" }));

}

}, [dispatch, genres, genresLoaded]);

onAuthStateChanged(firebaseAuth, (currentUser) => {

if (!currentUser) navigate("/login");

});

window.onscroll = () => {

setIsScrolled(window.pageYOffset === 0 ? false : true);

return () => (window.onscroll = null);

};

return (

<Container>

<Navbar isScrolled={isScrolled} />

<div className="hero">

<img

src={backgroundImage}

alt="background"

className="background-image"

/>

<div className="container">

<div className="logo">

<img src={MovieLogo} alt="Movie Logo" />

</div>

<div className="buttons flex">

<button

onClick={() => navigate("/player")}

className="flex j-center a-center"

>

<FaPlay />

Play

</button>

<button className="flex j-center a-center">

<AiOutlineInfoCircle />

More Info

</button>

</div>

</div>

</div>

<Slider movies={movies} />

</Container>

);

}

const Container = styled.div`

background-color: black;

.hero {

position: relative;

.background-image {

filter: brightness(60%);

}

img {

height: 100vh;

width: 100vw;

}

.container {

position: absolute;

bottom: 5rem;

.logo {

img {

width: 100%;

height: 100%;

margin-left: 5rem;

}

}

.buttons {

margin: 5rem;

gap: 2rem;

button {

font-size: 1.4rem;

gap: 1rem;

border-radius: 0.2rem;

padding: 0.5rem;

padding-left: 2rem;

padding-right: 2.4rem;

border: none;

cursor: pointer;

transition: 0.2s ease-in-out;

&:hover {

opacity: 0.8;

}

&:nth-of-type(2) {

background-color: rgba(109, 109, 110, 0.7);

color: white;

svg {

font-size: 1.8rem;

}

}

}

}

}

}

`;

export default Netflix;

- Code file Player.jsx:

import React from "react";

import styled from "styled-components";

import { BsArrowLeft } from "react-icons/bs";

import { useNavigate } from "react-router-dom";

import video from "../assets/video.mp4";

export default function Player() {

const navigate = useNavigate();

return (

<Container>

<div className="player">

<div className="back">

<BsArrowLeft onClick={() => navigate(-1)} />

</div>

<video src={video} autoPlay loop controls muted />

</div>

</Container>

);

}

const Container = styled.div`

.player {

width: 100vw;

height: 100vh;

.back {

position: absolute;

padding: 2rem;

z-index: 1;

svg {

font-size: 3rem;

cursor: pointer;

}

}

video {

height: 100%;

width: 100%;

object-fit: cover;

}

}

`;

- Code file Signup.jsx:

import {

createUserWithEmailAndPassword,

onAuthStateChanged,

} from "firebase/auth";

import React, { useState } from "react";

import { useNavigate } from "react-router-dom";

import styled from "styled-components";

import BackgroundImage from "../components/BackgroundImage";

import Header from "../components/Header";

import { firebaseAuth } from "../utils/firebase-config";

function Signup() {

const [showPassword, setShowPassword] = useState(false);

const [formValues, setFormValues] = useState({

email: "",

password: "",

});

const navigate = useNavigate();

const handleSignIn = async () => {

try {

const { email, password } = formValues;

await createUserWithEmailAndPassword(firebaseAuth, email, password);

} catch (error) {

console.log(error);

}

};

onAuthStateChanged(firebaseAuth, (currentUser) => {

if (currentUser) navigate("/");

});

return (

<Container showPassword={showPassword}>

<BackgroundImage />

<div className="content">

<Header login />

<div className="body flex column a-center j-center">

<div className="text flex column">

<h1>Unlimited movies, TV shows and more.</h1>

<h4>Watch anywhere. Cancel anytime.</h4>

<h6>

Ready to watch? Enter your email to create or restart membership.

</h6>

</div>

<div className="form">

<input

type="email"

placeholder="Email address"

onChange={(e) =>

setFormValues({

...formValues,

[e.target.name]: e.target.value,

})

}

name="email"

value={formValues.email}

/>

{showPassword && (

<input

type="password"

placeholder="Password"

onChange={(e) =>

setFormValues({

...formValues,

[e.target.name]: e.target.value,

})

}

name="password"

value={formValues.password}

/>

)}

{!showPassword && (

<button onClick={() => setShowPassword(true)}>Get Started</button>

)}

</div>

{showPassword && <button onClick={handleSignIn}>Log In</button>}

</div>

</div>

</Container>

);

}

const Container = styled.div`

position: relative;

.content {

position: absolute;

top: 0;

left: 0;

background-color: rgba(0, 0, 0, 0.5);

height: 100vh;

width: 100vw;

display: grid;

grid-template-rows: 15vh 85vh;

.body {

gap: 1rem;

.text {

gap: 1rem;

text-align: center;

font-size: 2rem;

h1 {

padding: 0 25rem;

}

}

.form {

display: grid;

grid-template-columns: ${({ showPassword }) =>

showPassword ? "1fr 1fr" : "2fr 1fr"};

width: 60%;

input {

color: black;

border: none;

padding: 1.5rem;

font-size: 1.2rem;

border: 1px solid black;

&:focus {

outline: none;

}

}

button {

padding: 0.5rem 1rem;

background-color: #e50914;

border: none;

cursor: pointer;

color: white;

font-weight: bolder;

font-size: 1.05rem;

}

}

button {

padding: 0.5rem 1rem;

background-color: #e50914;

border: none;

cursor: pointer;

color: white;

border-radius: 0.2rem;

font-weight: bolder;

font-size: 1.05rem;

}

}

}

`;

export default Signup;

- Code file TVShows.jsx:

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

import styled from "styled-components";

import Navbar from "../components/Navbar";

import { onAuthStateChanged } from "firebase/auth";

import { firebaseAuth } from "../utils/firebase-config";

import { useNavigate } from "react-router-dom";

import { useSelector, useDispatch } from "react-redux";

import { fetchMovies, getGenres } from "../store";

import SelectGenre from "../components/SelectGenre";

import Slider from "../components/Slider";

function TVShows() {

const [isScrolled, setIsScrolled] = useState(false);

const movies = useSelector((state) => state.netflix.movies);

const genres = useSelector((state) => state.netflix.genres);

const genresLoaded = useSelector((state) => state.netflix.genresLoaded);

const navigate = useNavigate();

const dispatch = useDispatch();

useEffect(() => {

if (!genres.length) dispatch(getGenres());

}, [dispatch, genres]);

useEffect(() => {

if (genresLoaded) {

dispatch(fetchMovies({ genres, type: "tv" }));

}

}, [dispatch, genres, genresLoaded]);

const [ setUser] = useState(undefined);

onAuthStateChanged(firebaseAuth, (currentUser) => {

if (currentUser) setUser(currentUser.uid);

else navigate("/login");

});

window.onscroll = () => {

setIsScrolled(window.pageYOffset === 0 ? false : true);

return () => (window.onscroll = null);

};

return (

<Container>

<Navbar isScrolled={isScrolled} />

<div className="data">

<SelectGenre genres={genres} type="tv" />

{movies.length ? (

<>

<Slider movies={movies} />

</>

) : (

<h1 className="not-available">

No TV Shows avaialble for the selected genre. Please select a

different genre.

</h1>

)}

</div>

</Container>

);

}

const Container = styled.div`

.data {

margin-top: 8rem;

.not-available {

text-align: center;

margin-top: 4rem;

}

}

`;

export default TVShows;

- Code file BackgroungImage.jsx:

import React from "react";

import styled from "styled-components";

import background from "../assets/login.jpg";

export default function BackgroundImage() {

return (

<Container>

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

</Container>

);

}

const Container = styled.div`

height: 100vh;

width: 100vw;

img {

height: 100vh;

width: 100vw;

}

`;

- Code file Card.jsx:

import React, { useState } from "react";

import { useNavigate } from "react-router-dom";

import styled from "styled-components";

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

import { RiThumbUpFill, RiThumbDownFill } from "react-icons/ri";

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

import video from "../assets/video.mp4";

export default React.memo(function Card({ index, movieData, isLiked = false }) {

const navigate = useNavigate();

const [isHovered, setIsHovered] = useState(false);

return (

<Container

onMouseEnter={() => setIsHovered(true)}

onMouseLeave={() => setIsHovered(false)}

>

<img

src={`https://image.tmdb.org/t/p/w500${movieData.image}`}

alt="card"

onClick={() => navigate("/player")}

/>

{isHovered && (

<div className="hover">

<div className="image-video-container">

<img

src={`https://image.tmdb.org/t/p/w500${movieData.image}`}

alt="card"

onClick={() => navigate("/player")}

/>

<video

src={video}

autoPlay={true}

loop

muted

onClick={() => navigate("/player")}

/>

</div>

<div className="info-container flex column">

<h3 className="name" onClick={() => navigate("/player")}>

{movieData.name}

</h3>

<div className="icons flex j-between">

<div className="controls flex">

<IoPlayCircleSharp

title="Play"

onClick={() => navigate("/player")}

/>

<RiThumbUpFill title="Like" />

<RiThumbDownFill title="Dislike"/>

</div>

<div className="info">

<BiChevronDown title="More Info"/>

</div>

</div>

<div className="genres flex">

<ul className="flex">

{movieData.genres.map((genre) => (

<li>{genre}</li>

))}

</ul>

</div>

</div>

</div>

)}

</Container>

);

});

const Container = styled.div`

max-width: 230px;

width: 230px;

height: 100%;

cursor: pointer;

position: relative;

img {

border-radius: 0.2rem;

width: 100%;

height: 100%;

z-index: 10;

}

.hover {

z-index: 99;

height: max-content;

width: 20rem;

position: absolute;

top: -18vh;

left: 0;

border-radius: 0.3rem;

box-shadow: rgba(0, 0, 0, 0.75) 0px 3px 10px;

background-color: #181818;

transition: 0.3s ease-in-out;

.image-video-container {

position: relative;

height: 140px;

img {

width: 100%;

height: 140px;

object-fit: cover;

border-radius: 0.3rem;

top: 0;

z-index: 4;

position: absolute;

}

video {

width: 100%;

height: 140px;

object-fit: cover;

border-radius: 0.3rem;

top: 0;

z-index: 5;

position: absolute;

}

}

.info-container {

padding: 1rem;

gap: 0.5rem;

}

.icons {

.controls {

display: flex;

gap: 1rem;

}

svg {

font-size: 2rem;

cursor: pointer;

transition: 0.3s ease-in-out;

&:hover {

color: #b8b8b8;

}

}

}

.genres {

ul {

gap: 1rem;

li {

padding-right: 0.7rem;

&:first-of-type {

list-style-type: none;

}

}

}

}

}

`;

- Code file CardSlider.jsx:

import React, { useRef, useState } from "react";

import styled from "styled-components";

import { AiOutlineLeft, AiOutlineRight } from "react-icons/ai";

import Card from "./Card";

export default React.memo(function CardSlider({ data, title }) {

const listRef = useRef();

const [sliderPosition, setSliderPosition] = useState(0);

const [showControls, setShowControls] = useState(false);

const handleDirection = (direction) => {

let distance = listRef.current.getBoundingClientRect().x - 70;

if (direction === "left" && sliderPosition > 0) {

listRef.current.style.transform = `translateX(${230 + distance}px)`;

setSliderPosition(sliderPosition - 1);

}

if (direction === "right" && sliderPosition < 4) {

listRef.current.style.transform = `translateX(${-230 + distance}px)`;

setSliderPosition(sliderPosition + 1);

}

};

return (

<Container

className="flex column"

showControls={showControls}

onMouseEnter={() => setShowControls(true)}

onMouseLeave={() => setShowControls(false)}

>

<h1>{title}</h1>

<div className="wrapper">

<div

className={`slider-action left ${

!showControls ? "none" : ""

} flex j-center a-center`}

>

<AiOutlineLeft onClick={() => handleDirection("left")} />

</div>

<div className="slider flex" ref={listRef}>

{data.map((movie, index) => {

return <Card movieData={movie} index={index} key={movie.id} />;

})}

</div>

<div

className={`slider-action right ${

!showControls ? "none" : ""

} flex j-center a-center`}

>

<AiOutlineRight onClick={() => handleDirection("right")} />

</div>

</div>

</Container>

);

});

const Container = styled.div`

gap: 1rem;

position: relative;

padding: 2rem 0;

h1 {

margin-left: 50px;

}

.wrapper {

.slider {

width: max-content;

gap: 1rem;

transform: translateX(0px);

transition: 0.3s ease-in-out;

margin-left: 50px;

}

.slider-action {

position: absolute;

z-index: 99;

height: 100%;

top: 0;

bottom: 0;

width: 50px;

transition: 0.3s ease-in-out;

svg {

font-size: 2rem;

}

}

.none {

display: none;

}

.left {

left: 0;

}

.right {

right: 0;

}

}

`;

- Code file Header.jsx:

import React from "react";

import { useNavigate } from "react-router-dom";

import styled from "styled-components";

import logo from "../assets/logo.png";

export default function Header(props) {

const navigate = useNavigate();

return (

<StyledHeader className="flex a-center j-between">

<div className="logo">

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

</div>

<button onClick={() => navigate(props.login ? "/login" : "/signup")}>

{props.login ? "Log In" : "Sign Up"}

</button>

</StyledHeader>

);

}

const StyledHeader = styled.header`

padding: 0 4rem;

.logo {

img {

height: 5rem;

}

}

button {

padding: 0.5rem 1rem;

background-color: #e50914;

border: none;

cursor: pointer;

color: white;

border-radius: 0.2rem;

font-weight: bolder;

font-size: 1.05rem;

}

`;

- Code file Navbar.jsx:

import { signOut } from "firebase/auth";

import React, { useState } from "react";

import { Link } from "react-router-dom";

import styled from "styled-components";

import logo from "../assets/logo.png";

import { firebaseAuth } from "../utils/firebase-config";

import { FaPowerOff, FaSearch } from "react-icons/fa";

export default function Navbar({ isScrolled }) {

const [showSearch, setShowSearch] = useState(false);

const [inputHover, setInputHover] = useState(false);

const links = [

{ name: "Home", link: "/" },

{ name: "TV Shows", link: "/tv" },

{ name: "Movies", link: "/movies" },

];

return (

<Container>

<nav className={`${isScrolled ? "scrolled" : ""} flex`}>

<div className="left flex a-center">

<div className="brand flex a-center j-center">

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

</div>

<ul className="links flex">

{links.map(({ name, link }) => {

return (

<li key={name}>

<Link to={link}>{name}</Link>

</li>

);

})}

</ul>

</div>

<div className="right flex a-center">

<div className={`search ${showSearch ? "show-search" : ""}`}>

<button

onFocus={() => setShowSearch(true)}

onBlur={() => {

if (!inputHover) {

setShowSearch(false);

}

}}

>

<FaSearch />

</button>

<input

type="text"

placeholder="Search"

onMouseEnter={() => setInputHover(true)}

onMouseLeave={() => setInputHover(false)}

onBlur={() => {

setShowSearch(false);

setInputHover(false);

}}

/>

</div>

<button onClick={() => signOut(firebaseAuth)}>

<FaPowerOff />

</button>

</div>

</nav>

</Container>

);

}

const Container = styled.div`

.scrolled {

background-color: black;

}

nav {

position: sticky;

top: 0;

height: 6.5rem;

width: 100%;

justify-content: space-between;

position: fixed;

top: 0;

z-index: 2;

padding: 0 4rem;

align-items: center;

transition: 0.3s ease-in-out;

.left {

gap: 2rem;

.brand {

img {

height: 4rem;

}

}

.links {

list-style-type: none;

gap: 2rem;

li {

a {

color: white;

text-decoration: none;

}

}

}

}

.right {

gap: 1rem;

button {

background-color: transparent;

border: none;

cursor: pointer;

&:focus {

outline: none;

}

svg {

color: #f34242;

font-size: 1.2rem;

}

}

.search {

display: flex;

gap: 0.4rem;

align-items: center;

justify-content: center;

padding: 0.2rem;

padding-left: 0.5rem;

button {

background-color: transparent;

border: none;

&:focus {

outline: none;

}

svg {

color: white;

font-size: 1.2rem;

}

}

input {

width: 0;

opacity: 0;

visibility: hidden;

transition: 0.3s ease-in-out;

background-color: transparent;

border: none;

color: white;

&:focus {

outline: none;

}

}

}

.show-search {

border: 1px solid white;

background-color: rgba(0, 0, 0, 0.6);

input {

width: 100%;

opacity: 1;

visibility: visible;

padding: 0.3rem;

}

}

}

}

`;

- Code file NotAvailable.jsx:

import React from "react";

export default function NotAvailable() {

return (

<h1 className="not-available">

No Movies avaialble for the selected genre. Please select a different

genre.

</h1>

);

}

# **KẾT LUẬN**

Dự án này đã đem lại cho em nhiều kiến thức và kinh nghiệm về lập trình sử dụng API, đặc biệt là trong việc xây dựng một trang web review phim. Em đã học được cách sử dụng các công nghệ như React, Firebase, Axios, React Redux và các thư viện khác để xây dựng ứng dụng web hiệu quả.

Em cũng đã gặp phải một số thách thức trong quá trình phát triển dự án, bao gồm việc xử lý dữ liệu từ API, các chức năng đăng kí đăng nhập và thiết kế giao diện người dùng hấp dẫn và thân thiện với người dùng.

Tuy nhiên, em đã đưa ra các giải pháp và kết quả cuối cùng là một ứng dụng web đáp ứng được các yêu cầu cơ bản của đề tài, bao gồm chức năng đăng ký, đăng nhập, lọc phim và TV Show theo thể loại, cùng với việc hiển thị các bộ phim và TV Show.

Với sự phát triển của công nghệ và nhu cầu sử dụng Internet ngày càng tăng, việc xây dựng một trang web review phim đáp ứng được nhu cầu của người dùng là rất cần thiết và có tiềm năng phát triển. Em hy vọng rằng dự án này sẽ có ích cho các nhà phát triển và người dùng trong tương lai.

# **HƯỚNG PHÁT TRIỂN**

Để phát triển tiếp dự án này, có thể cải thiện và phát triển nhiều tính năng hữu ích hơn với các tính năng như:

* Tính năng tìm kiếm: Hiện tại trang web chưa tìm kiếm được, có thể thêm chức năng tìm kiếm theo đạo diễn, diễn viên, năm sản xuất, thể loại phim, và các tiêu chí khác.
* Tính năng đánh giá và bình luận: Người dùng có thể đánh giá và viết nhận xét về các bộ phim mà họ đã xem, và các đánh giá này sẽ được hiển thị trên trang chi tiết của từng phim.
* Tính năng lưu phim: Người dùng có thể lưu phim vào danh sách yêu thích, và danh sách này sẽ được lưu trữ trên server để người dùng có thể truy cập từ bất cứ thiết bị nào.
* Cải thiện giao diện người dùng: Giao diện người dùng là yếu tố quan trọng giúp ứng dụng trở nên thu hút và dễ sử dụng. Có thể cải thiện giao diện người dùng bằng cách thay đổi màu sắc, font chữ, hoặc thay đổi vị trí các thành phần trong trang.
* Tăng cường tính bảo mật: Bảo mật là yếu tố quan trọng để đảm bảo an toàn cho người dùng. Có thể tăng cường tính bảo mật bằng cách thêm tính năng xác thực người dùng, mã hóa dữ liệu, hoặc thực hiện kiểm tra bảo mật thường xuyên để phát hiện và khắc phục các lỗ hổng bảo mật.

# **TÀI LIỆU THAM KHẢO**

* Tài liệu học phần Lập trình sử dụng API của thầy Đinh Công Tùng
* ChatGPT
* Firebase Doc
* The Movie Database API Doc
* Video trên Youtube