

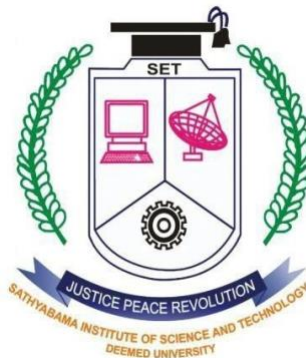
PROFESSIONAL TRAINING REPORT
At
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Submitted in partial fulfillment of the requirements for the
award of Bachelor of Engineering Degree in

Information Technology

By

ADHITHYA.V
(Reg.No.39120004)



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE
AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600119
www.sathyabamauniversity.ac.in

APRIL 2022



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade “A” by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI– 600119
www.sathyabama.ac.in

DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ADHITHYA.V**
(Reg. No: 39120004) who carried out the project entitled “**MOVIES AND TV SERIES APPLICATION**” under my supervision from March 2022 to April 2022.

Internal Guide

Dr. S. REVATHY , M.E., Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **ADHITHYA.V** (Reg.No.39120004) here by declare that the Project report entitled "**Movies And TV series Application**" done by me under the guidance of the **Dr.S.REVATHY,M.E.,Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA, M.E., Ph.D., Dean, School of Computing** and **Dr.R.SUBHASHINI, M.E.,Ph.D., Head of the Department, Dept.of Information Technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.S.REVATHY, M.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



ABSTRACT

New media is creating a new world and changing our daily lives. Internet is changing Indian cinema very rapidly. There are several new technologies, which are used by audience to watch movies and documentaries on their personal gadgets. Netflix, Amazon Prime, Hotstar, Zee5 and ALT Balaji, Jio tv are some examples, which are providing movies and other video content through internet. These platforms are new convergence of television and cinema. Some new ventures are producing video content for these Internet platforms only.

Now audience can watch new movies, web series, documentaries, news and short films on their convenient timing at reasonable price on these new media platforms. These new ventures are changing Indian television and cinema in many ways. Objectives of this study are to know how Indian Television and Cinema are changing due to new online platforms and how youth is using these new digital platforms for video content watching.

This study analyses the video content watching habits of the youth and try to find out the changes in television and cinema watching trends in youth. This study is based on survey method to know the video content watching habits of youths through online platforms. Study reveals that Hotstar, Netflix and Jio are the major players in Indian OTT service market. Indian audience use these applications without paying any money. Most of the viewers watch content through these applications up to 2 hours daily.

Most of the audience watch content on streaming media in night. Most preferred content on Over-the-top applications is web series. Indian population loves to watch web series on these platforms. Second preferred program is movie. Hindi is preferred language of Indian viewers. Entertainment is the biggest reason behind the use of over-the-top application. Most of the Indian users watch movies on these applications. Action and comedy are the favourite genre of movie. Indian audience loves to watch latest movies on these applications.

Almost all respondents agree that over the top applications are changing television and film watching habits in India. Reasons behind the change are convenience of service, personal medium and availability of international content. Study reveals that future of Over-the-top applications is bright in India and the reasons for this are smartphone penetration, international collaborations between media moguls, Cost effectiveness and digital quality of the medium.!

TABLES OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF FIGURES	ii
	LIST OF ABBREVIATIONS	iii
1.	INTRODUCTION	1
	1.1 Movie Web Application with ReactJS	1
	1.1.1 Creating the Project	1
	1.1.2 How to Set Up a React Project	1
	1.1.3 Material-UI	2
	1.1.4 Installation of Material-UI	2
	1.1.5 React Router DOM	2
	1.1.6 Virtual DOM	2
	1.2 Role of HTML, CSS, JAVASCRIPT, REACT JS	2
	1.3 Requirements for Applications	4
2.	MOVIES AND TV SERIES APPLICATION	5
	2.1 Aim	5
	2.2 Project Scope	5
3.	SYSTEM SPECIFICATION	6
	3.1 Hardware Requirements	6
	3.2 Software Requirements	6
	3.3 Technology I Used	6
	3.4 Benefits of using this Website	7
4.	SUMMARY AND CONCLUSIONS	8
	4.1 Conclusion	8
	4.2 Future Scope	8
	4.3 References	8
	4.4 Appendix	10
	4.4.1 Snapshots	10
	4.4.2 Source code	15

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	Home Page	10
1.2	Trending Page	11
1.3	Movies Page	11
1.4	Series Page	12
1.5	Search Page	13
1.6	Trailer Page	14

App	-	Application
HTML	-	Hyper Text Markup Language
CSS	-	Cascading Style Sheets
MUI	-	Material User Interface
JS	-	Java Script

1.INTRODUCTION

1.1 ReactJS:

ReactJS tutorial provides basic and advanced concepts of ReactJS. Currently, ReactJS is one of the most popular JavaScript front-end libraries which has a strong foundation and a large community.

ReactJS is a **declarative, efficient**, and flexible **JavaScript library** for building reusable UI components. It is an open-source, component-based front end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

We will be following the below steps for creating our application:

1.1.1 Creating the project:

We will start by creating a new react project using the create-react-app tool. You can use the following command for creating one with the desired name. Please ensure that Node and npm are installed on the system.!

1.1.2 How to Set Up a React Project:

Creating a React project. We're going to use create-react-app to get us off the ground quickly. Fire up a terminal and type: `npx create-react-app demo-app`. `cd demo-app`. When that's finished doing its thing, we're going to add Bootstrap to help us position things nicely without needing much of our own CSS. Run the following commands: `cd movie-app`, `npm install bootstrap`.

React Environment Setup

Pre-requisite for ReactJS

1. NodeJS and NPM
2. React and React DOM
3. Webpack
4. Babel

There are two ways to set up an environment for successful ReactJS application. They are

1. **Using the npm command** (npm install -g create-react-app)
2. **Using the create-react-app command** (create-react-app react-project)

1.1.3 Material-UI:

Material-UI is simply a library that allows us to import and use different components to create a user interface in our React applications.

This saves a significant amount of time since the developers do not need to write everything from scratch.

1.1.4 Installation of Material-UI:

Install Material-UI, the world's most popular React UI framework. Material-UI is available as a npm package.

1.1.4 Install npm:

To install and save in your package. Json dependencies, run:

```
// with npm      npm install @material-ui/core
```

1.1.5 React Router DOM:

The react-router-dom package contains bindings for using React router in web applications.

1.1.6 VIRTUAL DOM:

ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

1.2 Role of HTML, CSS, JAVASCRIPT, REACT JS:

- A webpage can contain headings, paragraphs, images, videos, and many other types of data. Front end developers use the HTML element to specify what kind of information each item on a webpage contains — for instance, the “p” HTML element indicates a paragraph.
- Developers also write HTML language code to specify how different items relate to one another in the page's overall structure or document structure.
- CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. ...
- CSS helps us to control the text color, font style, the spacing between paragraphs, sizing of columns, layout designs, and many more.

- CSS provides these capabilities through a technical model based on a hierarchical scope of effect, the separation of style from content, and a well-defined set of published standards.
- JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more.
- It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.
- React is a declarative, efficient, and flexible JavaScript library for building user interfaces. It lets you compose complex UIs from small and isolated pieces of code called “components”.
- The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM.
- We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

1.3 Requirements of Application:

- a) Create a workspace with VS Code
- b) Design the general layout with HTML & CSS
- c) Create the required components with React JS.

In React application, there are several files and folders in the root directory. Some of them are as follows:

1. **node modules:** It contains the React library and any other third party libraries needed.
2. **public:** It holds the public assets of the application. It contains the index.html where React will mount the application by default on the `<div id="root"></div>` element.
3. **src:** It contains the App.css, App.js, App.test.js, index.css, index.js, and serviceWorker.js files. Here, the App.js file always responsible for displaying the output screen in React.
4. **package-lock.json:** It is generated automatically for any operations where npm package modifies either the node_modules tree or package.json. It cannot be published. It will be ignored if it finds any other place rather than the top-level package.
5. **package.json:** It holds various metadata required for the project. It gives information to npm, which allows to identify the project as well as handle the project's dependencies.
6. **README.md:** It provides the documentation to read about React topics.

2. MOVIES AND TV SERIES APPLICATION

2.1 AIM:

The aim to create a Movies and TV Series application using ReactJS.

2.2 PROJECTSCOPE:

Web applications have become the primary necessity for modern consumers. They help our customers to get their necessary products or services in a short time. The incorporation of technology in our day-to-day services like paying bills to booking cabs involves complex procedures and innumerable iterations to serve customers efficiently. As on-demand apps offer the comfort to get products delivered at the doorsteps, people need not have to commute to buy food, groceries, clothes, and other commercial necessities. Technology has made it possible for people to get their necessities with a few simple taps and swipes on their smartphones.

3. SYSTEM SPECIFICATION

3.1 HARDWARE REQUIREMENT

- ➔ RAM: 4GB
- ➔ Processor Intel Core i3 and above

3.2 SOFTWARE REQUIREMENT

➔ Front-end: ReactJS and TypeScript

3.3 TECHNOLOGY USED

I have built this project using the following tools & techniques:

- React.JS
 - JavaScript.
 - Material-UI.
 - Styled-Components.
 - React Router.
 - Chakra-UI This Movies and TV Series application has following features in it:
-
- The first step is to code for the desired pages for the OTT platform i.e., Visual Studio Code.
 - In this website the home page contains the categories such as Login, Trending, Movies, TV series, Search
 - The user can login and personalize his/her genre of interest.
 - The rating is displayed for each movie to help the user watch the best movie.
 - Facility to scroll to 'n' number of pages and explore all the genres.
 - With a click on the caption of the page the user is easily guided back from the bottom of the page from the top.
 - Using the search option, the user can search a movie of any language and can view list of movies related to the search name.
- 6
- With this binge-watching app one can watch the trailer of any movie of their interest and decide if it is worth the watch.
 - This app helps a movie lover to know all the details of the movie.
 - The ratings displayed for each movie guides the user to watch the best one.
 - The code is done from the scratch and the layout is made using frontend languages.

3.4 BENEFITS OF USING THIS WEBSITE:

1) Creates space for community building.

- 2) Creates a platform for budding creators.
- 3) Various contents bring diversity in people by making them aware of different culture.

4. SUMMARY AND CONCLUSIONS

4.1 SUMMARY:

- With this binge-watching app one can watch the trailer of any movie of their interest and decide if it is worth the watch.
- This app helps a movie lover to know all the details of the movie.

- The ratings displayed for each movie guides the user to watch the best one.
- The code is done from the scratch and the layout is made using frontend languages.

4.2 CONCLUSIONS:

- The front-end languages such as HTML, CSS, React JS and Material UI are becoming one among the emerging languages.
- In today's world, the web development technology is an emerging technique which helps in many real time applications and saves people time and efforts.
- On this account this binge-watching app is developed to entertain the users.

4.3 FUTURE SCOPE:

Netflix, Inc. is one of the best examples regarding commercial applications for Video Streaming. With millions subscribing to the service all over the world, the company has found a way to capitalize its services using its title stock and outsourced infrastructure

4.4 REFERENCES

1. www.w3schools.com
2. www.tutorialpoint.com
3. Clever Programmer-YouTube

4.5 APPENDIX

A. SNAPSHOTS

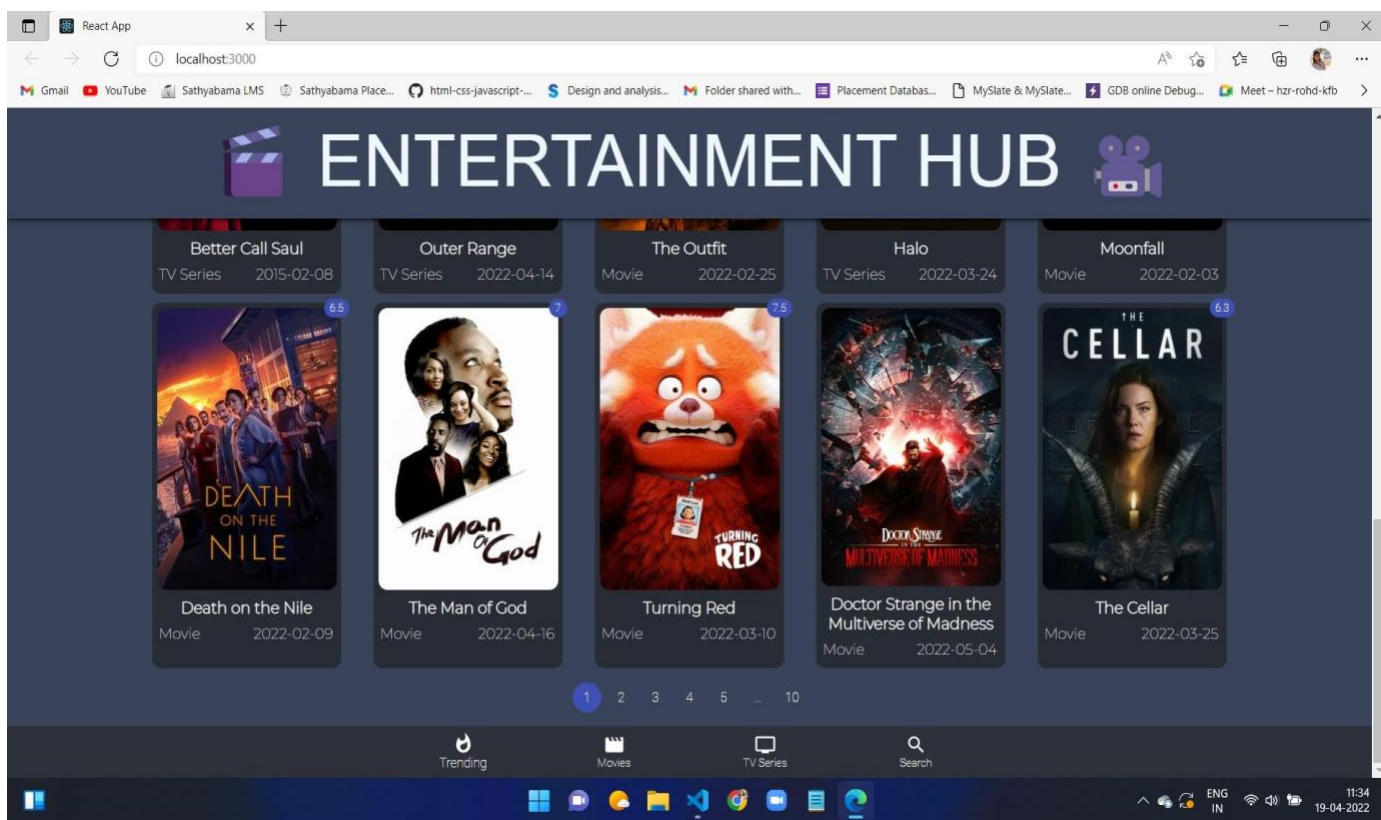


Fig 1.1: Home Page

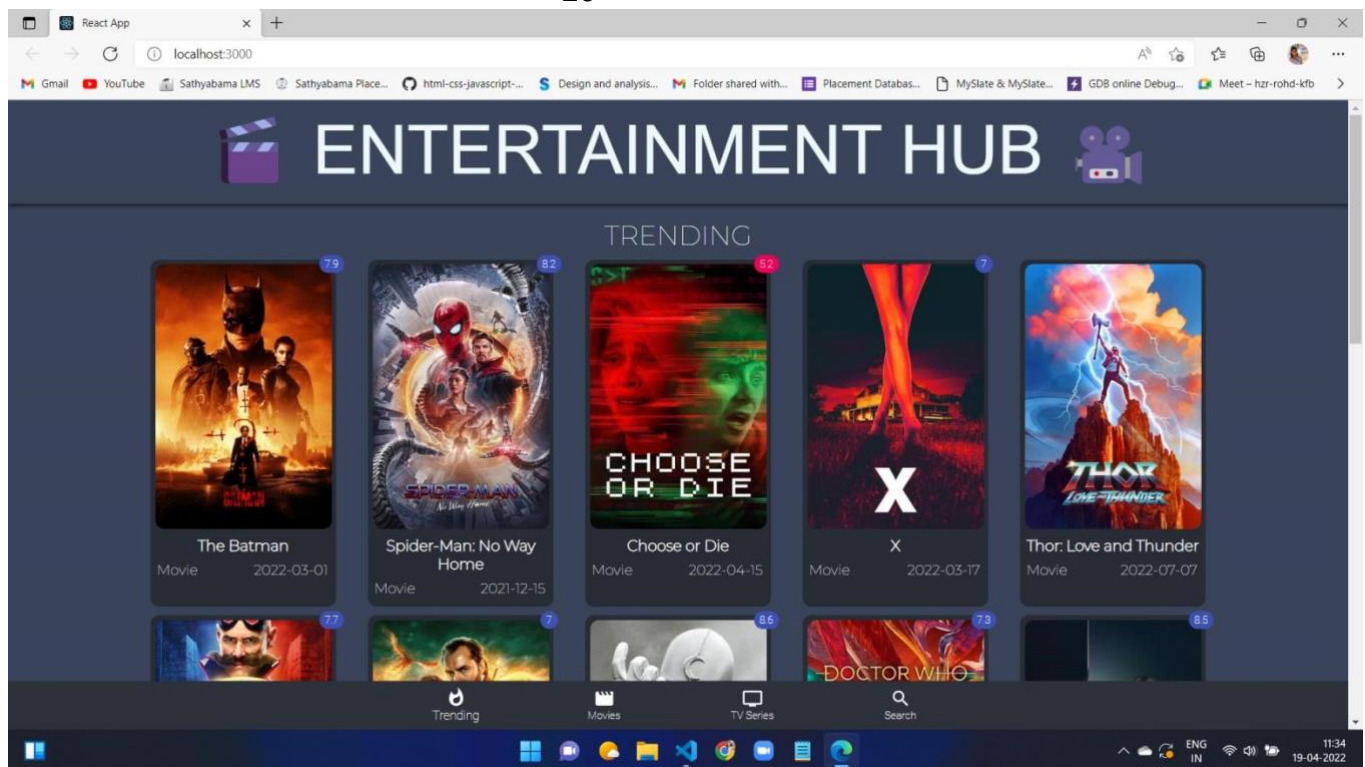


Fig 1.2: Trending Page

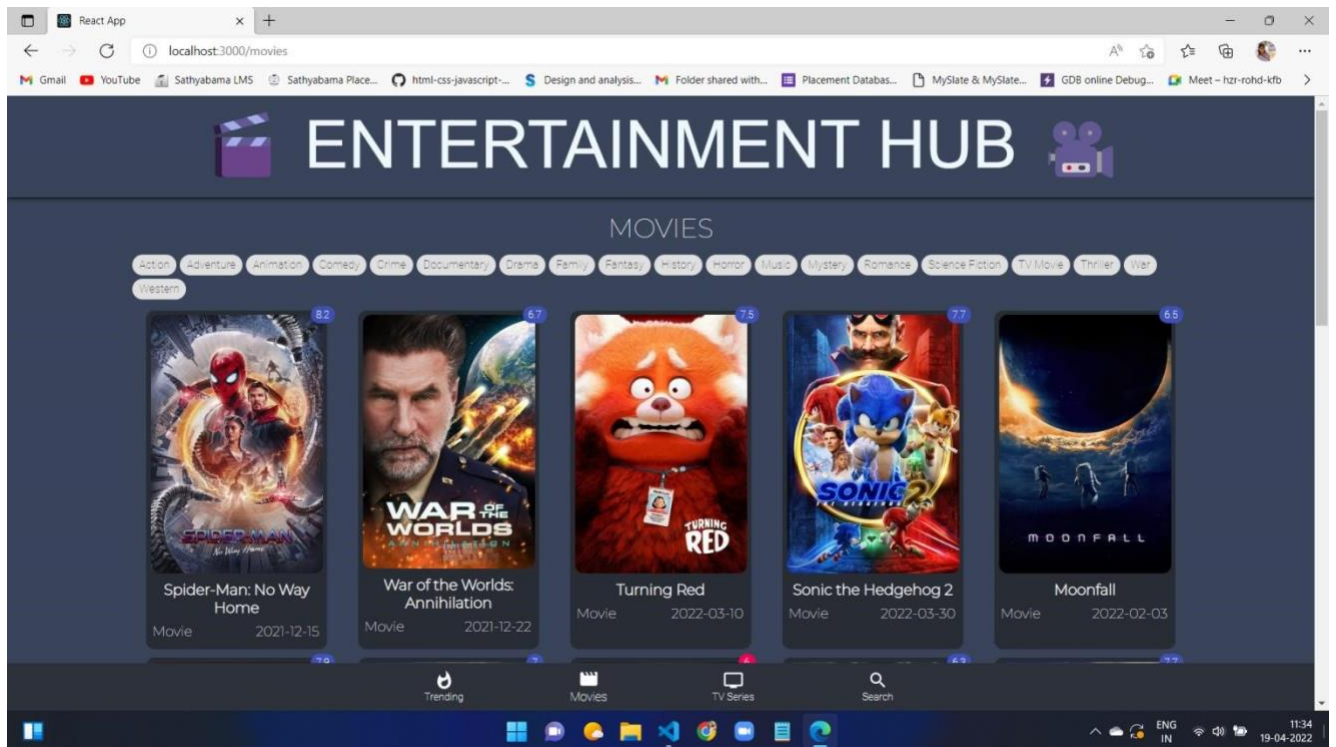


Fig 1.3: Movies Page

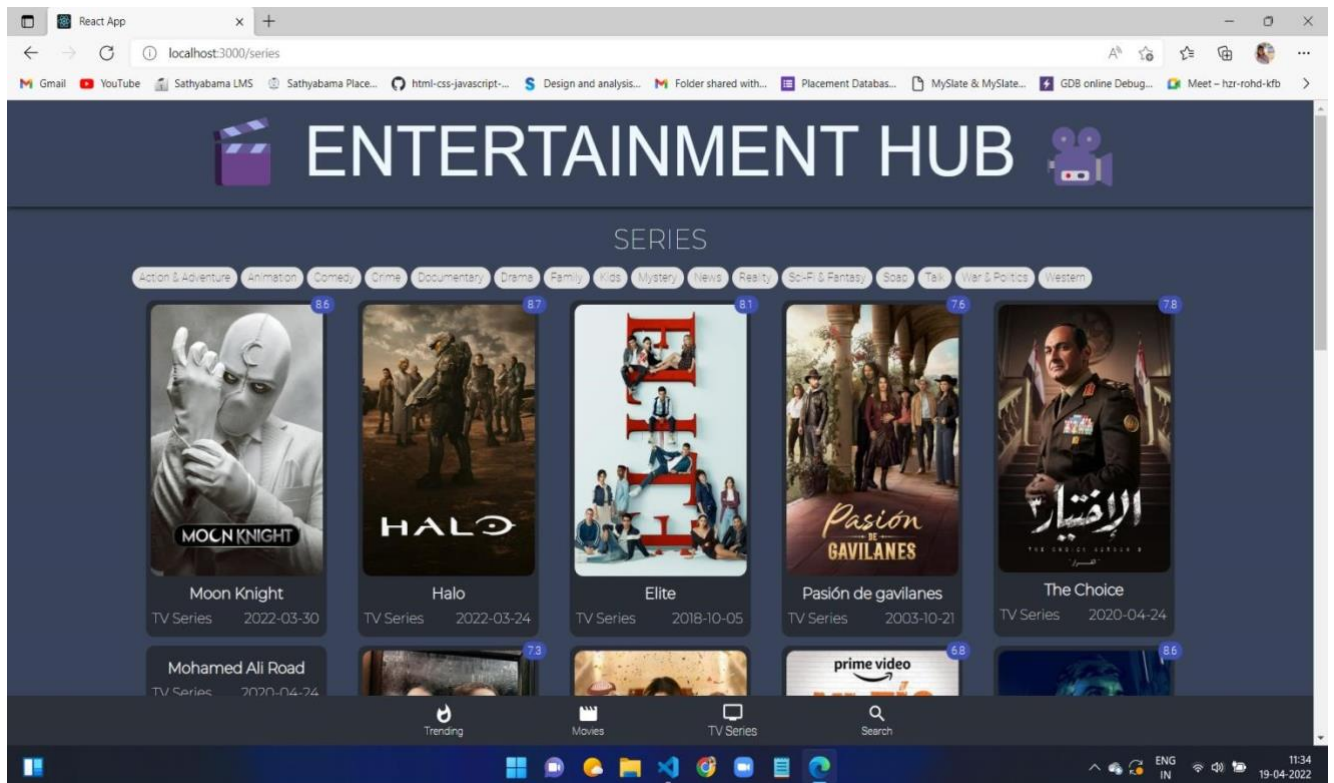


Fig 1.4: Series Page

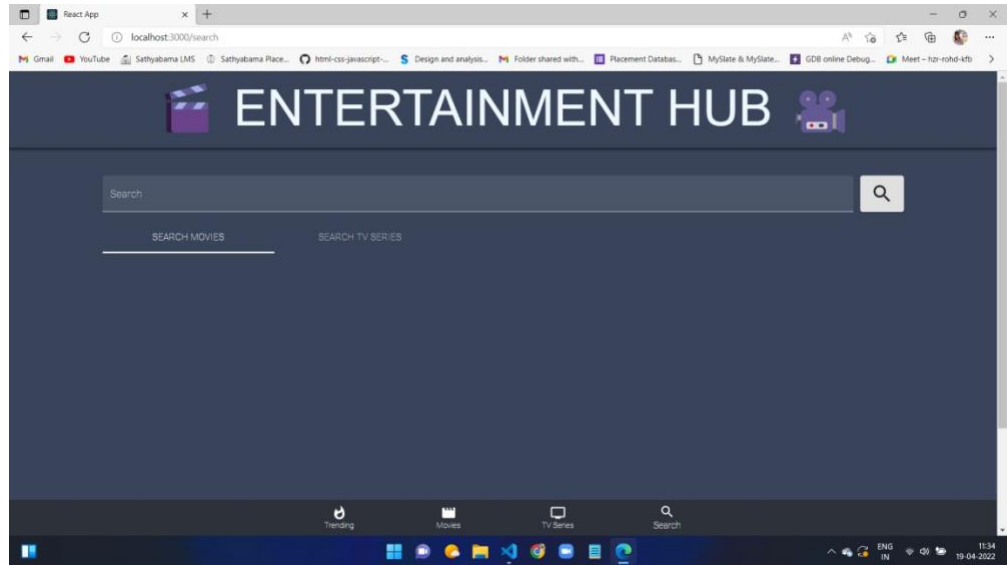
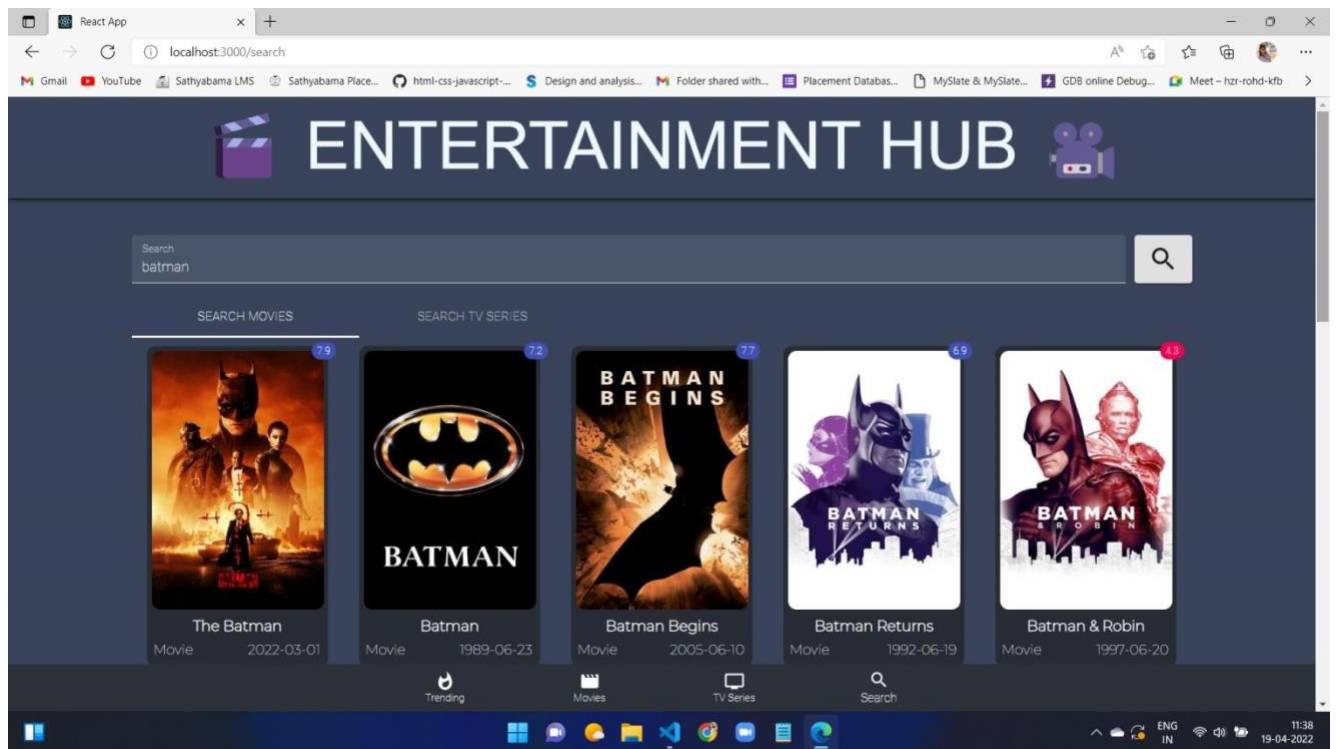


Fig 1.5: Search Page



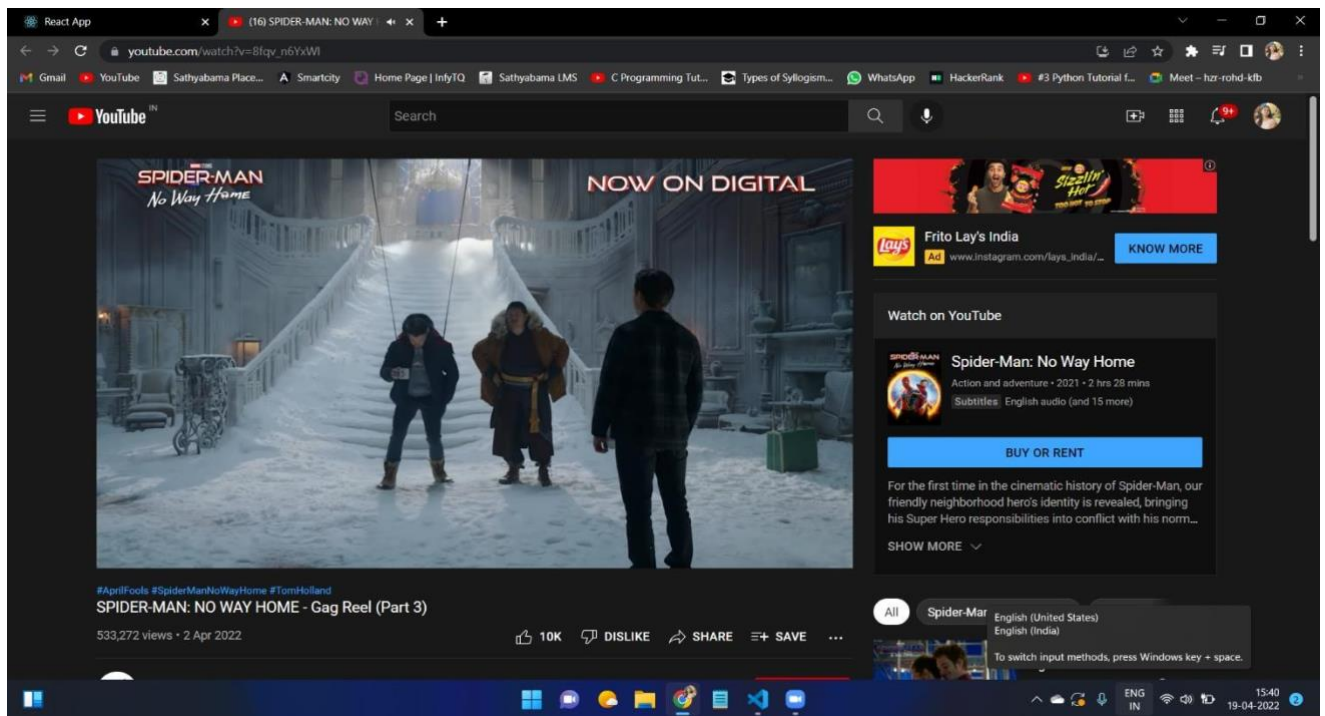
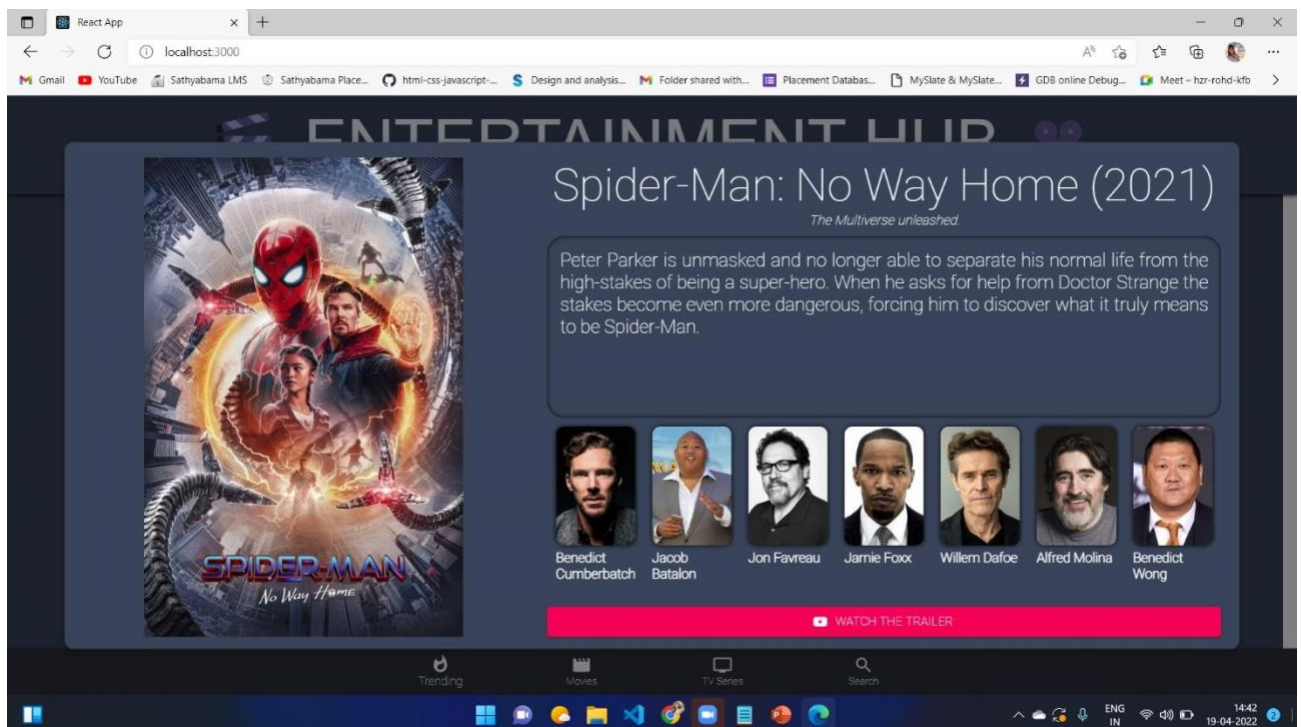


Fig 1.6 TRAILER PAGE



B. SOURCE CODE:

INDEX.JS:

```
import React from "react";
import ReactDOM from "react-dom"; import "./index.css";
import App from "./App";

ReactDOM.render(<App />, document.getElementById("root"));
```

APP.JS

```
APP.js import { Container } from "@material-ui/core"; import {
BrowserRouter, Route, Switch } from "react-router-dom"; import
'./App.css'; import Header from './components/Header'; import
SimpleBottomNavigation from './components/MainNav'; import
Trending from './components/Pages/Trending/Trending'; import
Movies from './components/Pages/Movies/Movies'; import Series
from './components/Pages/Series/Series'; import Search from
'./components/Pages/Search/Search';

function
App() {

  return (
    <BrowserRouter>
      <Header/>
      <div className="app">
        <Container>
          <Switch>
            <Route path="/" component={Trending} exact />
            <Route path='/movies' component={Movies}/>
            <Route path='/series' component={Series}/>
            <Route path='/search' component={Search}/>
          </Switch>
        </Container>
      </div>
    </BrowserRouter>
  );
}
```

```
        </Switch>

        </Container>
      </div>
      <SimpleBottomNavigation />
    </BrowserRouter>
  );
} export default
App;
```

MainNav.js

```
import React, { useEffect } from "react"; import { makeStyles } from
"@material-ui/core/styles"; import BottomNavigation from "@material-
ui/core/BottomNavigation"; import BottomNavigationAction from "@material-
ui/core/BottomNavigationAction"; import TvIcon from "@material-ui/icons/Tv";
import MovieIcon from "@material-ui/icons/Movie"; import SearchIcon from
"@material-ui/icons/Search"; import WhatshotIcon from "@material-
ui/icons/Whatshot"; import { useHistory } from "react-router-dom";

const useStyles =
makeStyles({  root: {
width: "100%",    position:
"fixed",    bottom: 0,
backgroundColor: "#2d313a",
zIndex: 100,
},
}); export default function
SimpleBottomNavigation() {  const classes =
useStyles();  const [value, setValue] =
React.useState(0);  const history = useHistory();
```

```

    useEffect(() => {      if
(value === 0) {
history.push("/");      }else
if (value === 1) {
history.push("/movies");
} else if (value === 2) {
history.push("/series");
} else if (value === 3 ) {
history.push("/search");
}
}, [value, history]);

    return (
      <BottomNavigation
value={value}      onChange={(event,
newValue) => {
setValue(newValue);
}}      showLabels
className={classes.root}
      >

        <BottomNavigationAction
style={{ color: "white" }}
label="Trending"
icon={<WhatshotIcon />}
        />
        <BottomNavigationAction
style={{ color: "white" }}
label="Movies"
icon={<MovieIcon />}
        />
        <BottomNavigationAction
style={{ color: "white" }}
label="TV Series"
icon={<TvIcon />}
        />
        <BottomNavigationAction
style={{ color: "white" }}
label="Search"
icon={<SearchIcon />}
        />
      </BottomNavigation>
    );
  }
}

```

Main Nav js

17 MOVIES.JS

```
import axios from "axios"; import { useEffect, useState } from
"react"; import Genres from ".././Genres"; import SingleContent
from ".././SingleContent/SingleContent"; import useGenre from
".././hooks/useGenre.js"; import CustomPagination from
".././Pagination/CustomPagination";
const Movies = () => {  const [genres, setGenres] =
useState([]);  const [selectedGenres, setSelectedGenres] =
useState([]);  const [page, setPage] = useState(1);
const [content, setContent] = useState([]);  const
[numOfPages, setNumOfPages] = useState();  const
genreforURL = useGenre(selectedGenres);
  // console.log(selectedGenres);
  const fetchMovies = async () => {
const { data } = await axios.get(
  `https://api.themoviedb.org/3/discover/movie?api_key=${process.env.REACT_APP_API_KEY}&language
=en-US&sort_by=popularity.desc&include_adult=false&include_video=false&page=${page}&with_genres=${genrefor
orURL}`  );  setContent(data.results);  setNumOfPages(data.total_pages);
  };
  useEffect(() => {
window.scroll(0, 0);
fetchMovies();
  // eslint-disable-next-line
  }, [genreforURL, page]);
  return
(
  <div>
    <span className="pageTitle">Movies</span>
    <Genres      type="movie"
selectedGenres={selectedGenres}
setSelectedGenres={setSelectedGenres}
genres={genres}
setGenres={setGenres}
setPage={setPage}
    />
```

```

    <div className="trending">
      {content && content.map((c) => (
<SingleContent
      key={c.id}
id={c.id}
      poster={c.poster_path}
title={c.title || c.name}
date={c.first_air_date || c.release_date}
media_type="movie"
vote_average={c.vote_average}
      />
    ))}
    </div>
    {numOfPages > 1 && (
      <CustomPagination setPage={setPage} numOfPages={numOfPages} />
    )}
  </div>
);
}; export default
Movies;

```

SEARCH.JS

```

import
{
  Button,
  createMuiTheme,
  Tab,
  Tabs,
  TextField,
  ThemeProvider,
} from "@material-ui/core"; import "./Search.css"; import
SearchIcon from "@material-ui/icons/Search"; import { useEffect,
useState } from "react"; import axios from "axios"; import
CustomPagination from "../../Pagination/CustomPagination"; import
SingleContent from "../../SingleContent/SingleContent";

```



```

const Search = () => {  const [type, setType] =
useState(0);  const [searchText, setSearchText] =
useState("");  const [page, setPage] =
useState(1);  const [content, setContent] =
useState([]);  const [numOfPages, setNumOfPages] =
useState();

  const darkTheme =
createMuiTheme({  palette: {
type: "dark",  primary: {
main: "#fff",
},
},
});  const fetchSearch = async ()
=> {  try {  const { data } =
await axios.get(
`https://api.themoviedb.org/3/search/${type ? "tv" : "movie"}?api_key=${
process.env.REACT_APP_API_KEY
}&language=en-US&query=${searchText}&page=${page}&include_adult=false`
);
setContent(data.results);
setNumOfPages(data.total_pages);
// console.log(data);
} catch (error) {
console.error(error);
}
};  useEffect(() =>
{  window.scrollTo(0,
0);  fetchSearch();
// eslint-disable-next-line
}, [type, page]);
return
(
  <div>
    <ThemeProvider theme={darkTheme}>
      <div style={{display:"flex", margin:"15px 0"}}></div>
      <div className="search">
        <TextField
          style={{ flex: 1 }}
          className="searchBox"
          label="Search"
          variant="filled"
          onChange={(e) =>
            setSearchText(e.target.value)}
        />
        <Button
          onClick={fetchSearch}
          variant="contained"
          style={{ marginLeft: 10 }}
        >
          <SearchIcon fontSize="large" />

```

```

</Button>
</div>      <Tabs
value={type}
indicatorColor="primary"
textColor="primary"
onChange={(event, newValue) => {
setType(newValue);
setPage(1);
}} style={{
paddingBottom: 5 }} aria-
label="disabled tabs example"
>
  <Tab style={{ width: "50%" }} label="Search Movies" />
  <Tab style={{ width: "50%" }} label="Search TV Series" />
</Tabs>
</ThemeProvider>
<div className="trending">
  {content && content.map((c) => (
<SingleContent
  key={c.id}
id={c.id}
poster={c.poster_path}
title={c.title || c.name}
date={c.first_air_date || c.release_date}
media_type={type ? "tv" : "movie"}
vote_average={c.vote_average}
/>
))}
  {searchText &&
!content &&
(type? <h2>No Series Found</h2> : <h2>No Movies Found</h2>)}
</div>
  {numOfPages > 1 && (
    <CustomPagination setPage={setPage} numOfPages={numOfPages} />
  )}
</div>
);
};
export default
Search;

```


21 SERIES.JS

```

import axios from "axios"; import { useEffect, useState } from
"react"; import Genres from "../..//Genres"; import
CustomPagination from "../..//Pagination/CustomPagination";
import SingleContent from "../..//SingleContent/SingleContent";
import useGenre from "../..//hooks/useGenre";
const Series = () => { const [genres, setGenres] =
useState([]); const [selectedGenres, setSelectedGenres] =
useState([]); const [page, setPage] = useState(1);
const [content, setContent] = useState([]); const
[numOfPages, setNumOfPages] = useState(); const
genreforURL = useGenre(selectedGenres);
const fetchSeries = async () =>
{ const { data } = await
axios.get(
`https://api.themoviedb.org/3/discover/tv?api_key=${process.env.REACT_APP_API_KEY}&language=en
-
US&sort_by=popularity.desc&include_adult=false&include_video=false&page=${page}&with_genres=${genrefor
orURL}` ); setContent(data.results); setNumOfPages(data.total_pages);
// console.log(data);
};
useEffect(() => {
window.scroll(0, 0);
fetchSeries();
// eslint-disable-next-line
}, [genreforURL, page]);
return
(
<div>
<span className="pageTitle">Series</span>
<Genres type="tv"
selectedGenres={selectedGenres}
setSelectedGenres={setSelectedGenres}
genres={genres}
setGenres={setGenres}
setPage={setPage}
/>
<div className="trending">
{content && content.map((c) => (
<SingleContent key={c.id}
id={c.id} poster={c.poster_path}
title={c.title || c.name}
date={c.first_air_date || c.release_date}
media_type="tv"

```

```
vote_average={c.vote_average}
    />
    )})
</div>
{numOfPages > 1 && (
    <CustomPagination setPage={setPage} numOfPages={numOfPages} />
)}
</div>
);
};
export default
Series;
```

TRENDING.JS

```
import axios from "axios"; import { useEffect, useState } from
"react"; import '../..//SingleContent/SingleContent' import
SingleContent from "../..//SingleContent/SingleContent"; import
'./Trending.css'; import CustomPagination from
"../..//Pagination/CustomPagination";
const Trending = () => { const [page,
setPage] = useState(1); const [content,
setContent] = useState([]);

const fetchTrending = async ()
=> {
  const { data } = await
axios.get(
  `https://api.themoviedb.org/3/trending/all/day?api_key=${process.env.REACT_APP_API_KEY}&page=$
{page}`
);
  console.log(data.results);
  setContent(data.results);
};
  useEffect(() => {
window.scroll(0, 0);
fetchTrending();
    // eslint-disable-next-line
  }, [page]);
  return
(
  <div>
    <span className='pageTitle'> Trending </span>
    <div className="trending">
      {content && content.map((c) =>
<SingleContent
      key={c.id}
id={c.id} poster={c.poster_path}
title={c.title || c.name}
date={c.first_air_date || c.release_date}
```

```

media_type={c.media_type}
vote_average={c.vote_average}
      />
    })

    </div>
    <CustomPagination setPage={setPage}/>
  </div>
);
};
export default
Trending;

```

GENRES.JS

```

import { Chip } from "@material-
ui/core"; import axios from "axios";
import { useEffect } from "react";
const Genres = ({ selectedGenres,
setSelectedGenres, genres, setGenres, type,
setPage, }) => { const handleAdd = (genre) => {
setSelectedGenres([...selectedGenres, genre]);
setGenres(genres.filter((g) => g.id !== genre.id));
setPage(1);
};
const handleRemove = (genre) => { setSelectedGenres(
selectedGenres.filter((selected) => selected.id !== genre.id)
); setGenres([...genres,
genre]); setPage(1);
};
const fetchGenres = async () => {
const { data } = await axios.get(
`https://api.themoviedb.org/3/genre/${type}/list?api_key=${process.env.REACT_APP_API_KEY}&lang
uage=en-US`
); setGenres(data.genres);
};

```

```

useEffect(() =>
{
  fetchGenres();
  return () => {
    setGenres({}); // unmounting
  };
  // eslint-disable-next-line
}, []);

return (
  <div style={{ padding: "6px 0" }}>
    {selectedGenres.map((genre) => (
      <Chip
        style={{ margin: 2 }}
        label={genre.name}
        key={genre.id}
        color="primary"
        clickable
        size="small"
        onDelete={() =>
          handleRemove(genre)}
        />
      )
    )}
    {genres.map((genre) => (
      <Chip
        style={{ margin:
2 }}
        label={genre.name}
        key={genre.id}
        clickable
        size="small"
        onClick={() =>
          handleAdd(genre)}
        />
      )
    )}
  </div>
);
}); export default
Genres;

```

CONFIG.JS

```
//image sizes for tmdb export const img_300 =  
"https://image.tmdb.org/t/p/w300"; export const img_500 =  
"https://image.tmdb.org/t/p/w500";  
// contentModal and singleContent  
export const unavailable =  
"https://www.movienewz.com/img/films/poster-holder.jpg";  
// contentModal export const  
unavailableLandscape =  
"https://user-images.githubusercontent.com/10515204/56117400-9a911800-5f85-11e9-878b-3f998609a6c8.jpg";
```

25

```
// For Carousel export const noPicture =  
"https://upload.wikimedia.org/wikipedia/en/6/60/No_Picture.jpg";
```

HEADER.JS

```
import  
"./Header.css";  
const Header = () =>  
{  
  return (  
    <span onClick={() => window.scroll(0, 0)} className="header">  
       Entertainment  Hub  
    </span>  
  );  
}; export default  
Header;
```

API KEY:

```
REACT_APP_API_KEY=a0b55c807b587a12c01056f49bdfe4e4
```

PACKAGE.JSON

```
{
  "name": "react_project",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@chakra-ui/core": "^0.8.0",
    "@emotion/core": "^10.3.1",
    "@emotion/styled": "^10.3.0",
    "@material-ui/core": "^4.12.3",
    "@material-ui/icons": "^4.11.2",
    "@material-ui/lab": "^4.0.0-alpha.60",
    "@mui/icons-material": "^5.5.1",
    "@testing-library/jest-dom": "^5.16.2",
    "@testing-library/react": "^12.1.4",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^0.26.1",
    "chakra-ui-simple-autocomplete": "^1.0.0",
    "emotion-theming": "^10.3.0",
    "react": "^17.0.2",
    "react-alice-carousel": "^2.5.1",
    "react-dom": "^17.0.2",
```

26

```
"react-router-dom": "^5.2.0",
    "react-scripts": "5.0.0",
    "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"
    ]
  }
}
```

CUSTOMPAGINATION.JS

```
import React from "react"; import Pagination from "@material-
ui/lab/Pagination"; import { createMuiTheme, ThemeProvider } from
"@material-ui/core";
const darkTheme =
createMuiTheme({ palette: {
type: "dark",
},
}); export default function CustomPagination({ setPage, numOfPages = 10
}) {
  // Scroll to top when page changes
  const handlePageChange = (page) => {
    setPage(page);    window.scrollTo(0,
0);
  };
  return (    <div
style={{      width: "100%",
display: "flex",
justifyContent: "center",
marginTop: 10,
    }}
    >
```

27

```
<ThemeProvider theme={darkTheme}>
  <Pagination      onChange={(e) =>
handlePageChange(e.target.textContent)}      count={numOfPages}
color="primary"      hideNextButton      hidePrevButton
/>
  </ThemeProvider>
</div>
);
}
```

USEGENRES.JS

```
const useGenre = (selectedGenres) => {  
if (selectedGenres.length < 1) return "";  
    const GenreIds = selectedGenres.map((g) => g.id);  
return GenreIds.reduce((acc, curr) => acc + "," + curr);  
};    export default useGenre;
```

CONTENTMODEL.JS

```
import React, { useEffect, useState } from "react";  
import { makeStyles } from "@material-  
ui/core/styles"; import Modal from "@material-  
ui/core/Modal"; import Backdrop from "@material-  
ui/core/Backdrop"; import Fade from "@material-  
ui/core/Fade"; import axios from "axios"; import {  
img_500, unavailable,  
    unavailableLandscape, } from  
"../../config/config"; import "../ContentModal.css"  
import { Button } from "@material-ui/core"; import  
YouTubeIcon from "@material-ui/icons/YouTube";  
import Carousel from "../Carousel/Carousel";  
    const useStyles = makeStyles((theme) =>  
({ modal: { display: "flex",  
alignItems: "center", justifyContent:  
"center",  
    },
```

```

    paper: {      width: "90%",
height: "80%",      backgroundColor:
"#39445a",      borderRadius: 10,
color: "white",      boxShadow:
theme.shadows[5],      padding:
theme.spacing(1, 1, 3),
    },
  }));
  export default function TransitionsModal({ children, media_type, id })
{
  const classes = useStyles();
  const [open, setOpen] =
useState(false);
  const [content, setContent] = useState();
  const
[video, setVideo] = useState();
  const handleOpen = () =>
{
    setOpen(true);
  };
  const handleClose = () =>
{
    setOpen(false);
  };
  const fetchData = async () => {
const { data } = await axios.get(
  `https://api.themoviedb.org/3/${media_type}/${id}?api_key=${process.env.REACT_APP_API_KEY}&lan
guage=en-US`
);
setContent(data);
  // console.log(data);
};
  const fetchVideo = async () => {
const { data } = await axios.get(
  `https://api.themoviedb.org/3/${media_type}/${id}/videos?api_key=${process.env.REACT_APP_API_K
EY}&language=en-US`
);
setVideo(data.results[0]?.key);
};
  useEffect(() => {
    fetchData();
    fetchVideo();
    // eslint-disable-next-line
  }, []);

  return (
    <>
      <div
        className="media"
        style={{
          cursor: "pointer"
        }}
        color="inherit"
        onClick={handleOpen}

```



```

    >
      {children}
    </div>
    <Modal
      aria-
labelledby="transition-modal-title"
      aria-
describedby="transition-modal-description"
      className={classes.modal}
      open={open}
onClose={handleClose}
      closeAfterTransition

BackdropComponent={Backdrop}
BackdropProps={{
      timeout:
500,
    }}
  >
    <Fade in={open}>
      {content && (
        <div className={classes.paper}>
          <div className="ContentModal">
            <img
src={
content.poster_path
          ? `${img_500}/${content.poster_path}`
            : unavailable
          }
alt={content.name || content.title}
className="ContentModal__portrait"
          />
            <img
src={
content.backdrop_path
          ? `${img_500}/${content.backdrop_path}`
            : unavailableLandscape
          }
alt={content.name || content.title}
className="ContentModal__landscape"
          />
            <div className="ContentModal__about">
              <span className="ContentModal__title">
                {content.name || content.title} (
                  {(
content.first_air_date ||
content.release_date ||
                    "-----"
                  ).substring(0, 4)}
                )
              </span>
              {content.tagline && (
                <i className="tagline">{content.tagline}</i>
              )}

              <span className="ContentModal__description">
                {content.overview}
              </span>
            </div>
          </div>
        </div>
      )}
    </Fade>
  </Modal>

```

```

<div>
    <Carousel id={id} media_type={media_type} />
</div>

    <Button
        variant="contained"
startIcon={<YouTubeIcon />}
        color="secondary"
target="__blank"
href={`https://www.youtube.com/watch?v=${video}`}
    >
        Watch the Trailer
    </Button>
</div>
</div>
</div>
    )}
</Fade>
</Modal>
</>
);
}

```

CAROUSEL.JS

```
import axios from "axios"; import React, { useEffect,
useState } from "react"; import AliceCarousel from
"react-alice-carousel"; import "react-alice-
carousel/lib/alice-carousel.css"; import { img_300,
noPicture } from "../../config/config"; import
"./Carousel.css";
const handleDragStart = (e) =>
e.preventDefault();
const Gallery = ({ id, media_type }) => {
const [credits, setCredits] = useState([]);
const items = credits.map((c) =>
(
<div className="carouselItem">
<img
src={c.profile_path ? `${img_300}/${c.profile_path}` :
noPicture} alt={c?.name} onDragStart={handleDragStart}
className="carouselItem__img"
/>
<b className="carouselItem__txt">{c?.name}</b>
</div>
)); const
responsive = {
0: {
items: 3,
},
},
```

```

512:
{
    items: 5,
},
1024: {
items: 7,
},
};
const fetchCredits = async () =>
{
    const { data } = await
axios.get(
    `https://api.themoviedb.org/3/\${media\_type}/\${id}/credits?api\_key=\${process.env.REACT\_APP\_API\_KEY}&language=en-US`
    );
setCredits(data.cast);
};
useEffect(() =>
{
fetchCredits();
    // eslint-disable-next-line
}, []);

    return (
        <AliceCarousel
mouseTracking          infinite
disableDotsControls
disableButtonsControls
responsive={responsive}
items={items}          autoPlay
        />
    );
}; export default
Gallery;

```

APP.CSS

```
* { scroll-behavior:
smooth;
}
.app { min-height:
100vh; background-color:
#39445a; color: white;
padding-top: 130px;
padding-bottom: 70px;
}
@media (max-width: 700px) {
.app { padding-
top: 70px;
}
}
```

32

```
.pageTitle { text-transform:
uppercase; display: flex; justify-
content: center; font-family:
"Montserrat", sans-serif; font-size:
2vw; padding: 4px; border-radius:
50px; color: white;
}
@media (max-width: 1000px) {
.pageTitle { font-
size: 6.4vw;
}
}
```

