

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

KISHORE D
(Reg.No.39120055)



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 **KISHORE D (Reg. No: 39120055)** who carried out the project entitled “**MOVIES AND TV SERIES APPLICATION**” under my supervision from March 2022 to April 2022.

Internal Guide

Dr. R. SUBHASHINI, 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, **Kishore. D** (Reg.No.39120055) here by declare that the Project report entitled "**Movies And TV series Application**" done by me under the guidance of the **Dr. R. SUBHASHINI, 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. R. SUBHASHINI, 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 modern 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 analyzes 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 favorite 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 Movie Web Application with ReactJS Creating the Project How to set Up a React Project Role of CSS,JAVASCRIPT,REACT JS Requirements for Applications	1 1 1 2 4
2.	MOVIES AND TV SERIES APPLICATION Aim Project Scope	 5 5
3.	SYSTEM SPECIFICATION Hardware Requirements Software Requirements Technology I used Benefits of using this website	6 6 6 6

		7
4	SUMMARY AND CONCLUSION Conclusion Future Scope Reference Appendix Snapshots Source Code	8

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	Top Rated	12
1.5	Horror Movies	13

LISTOFABBREVIATIONS

App	-	Application
CSS	-	Cascading Style Sheets
JS	-	Java Script

1.INTRODUCTION

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:

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.!

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 movie-app`. `cd movie-app`. Run the following commands to get movie trailer `npm i react-youtube`, `npm i movie-trailer`.

React Environment Setup

Pre-requisite for ReactJS

1. NodeJS and NPM
2. React and React DOM

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)

Role of CSS, JAVASCRIPT, REACT JS:

- 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.

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.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.

2. MOVIES AND TV SERIES APPLICATION

AIM:

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

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

HARDWARE REQUIREMENT

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

SOFTWARE REQUIREMENT

- ☐ Front-end: ReactJS and TypeScript

TECHNOLOGY USED

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

- React.JS
- JavaScript.
- CSS

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.

- 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.

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

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.

CONCLUSIONS:

- The front-end languages such as CSS, React JS and MaterialUI 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.

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

REFERENCES

1. Clever Programmer-YouTube
2. Sonny Sangha-YouTube

APPENDIX

A. SNAPSHOTS

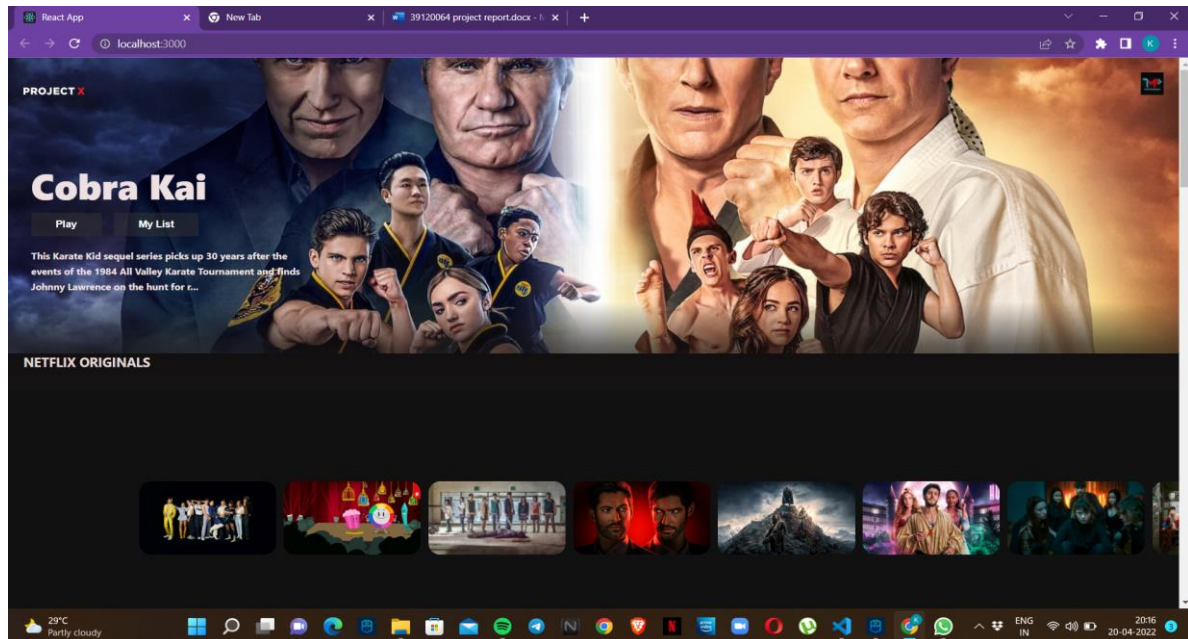


Fig 1.1: Home Page

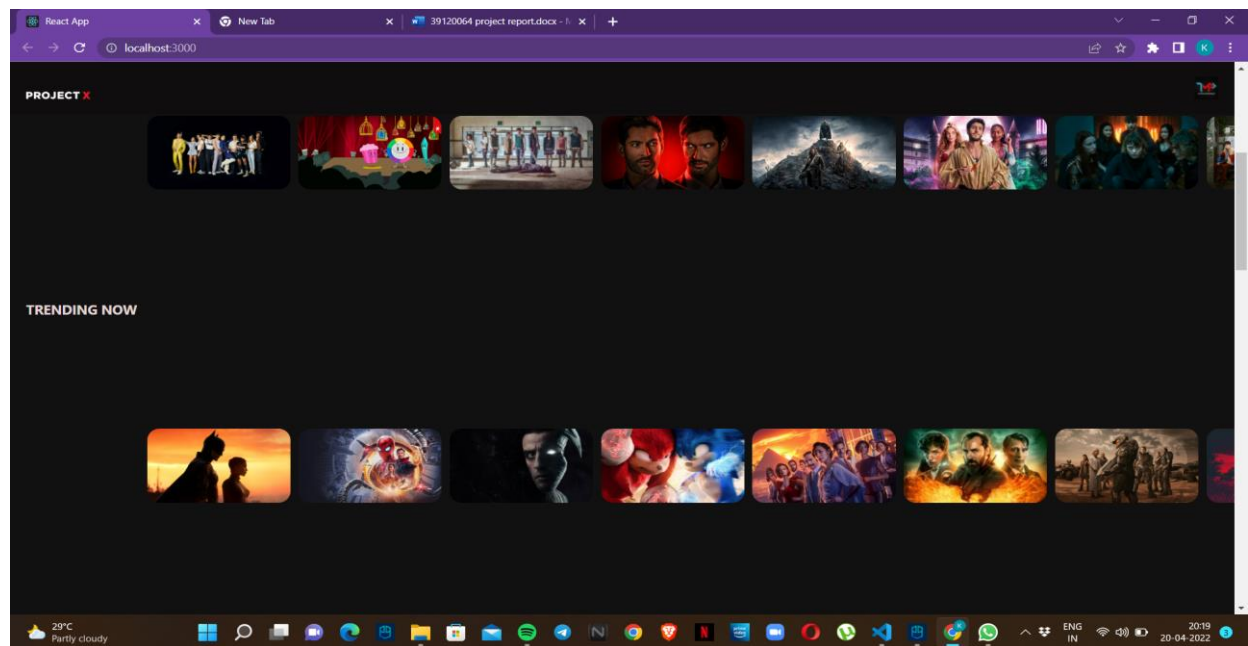


Fig 1.2: Trending Page

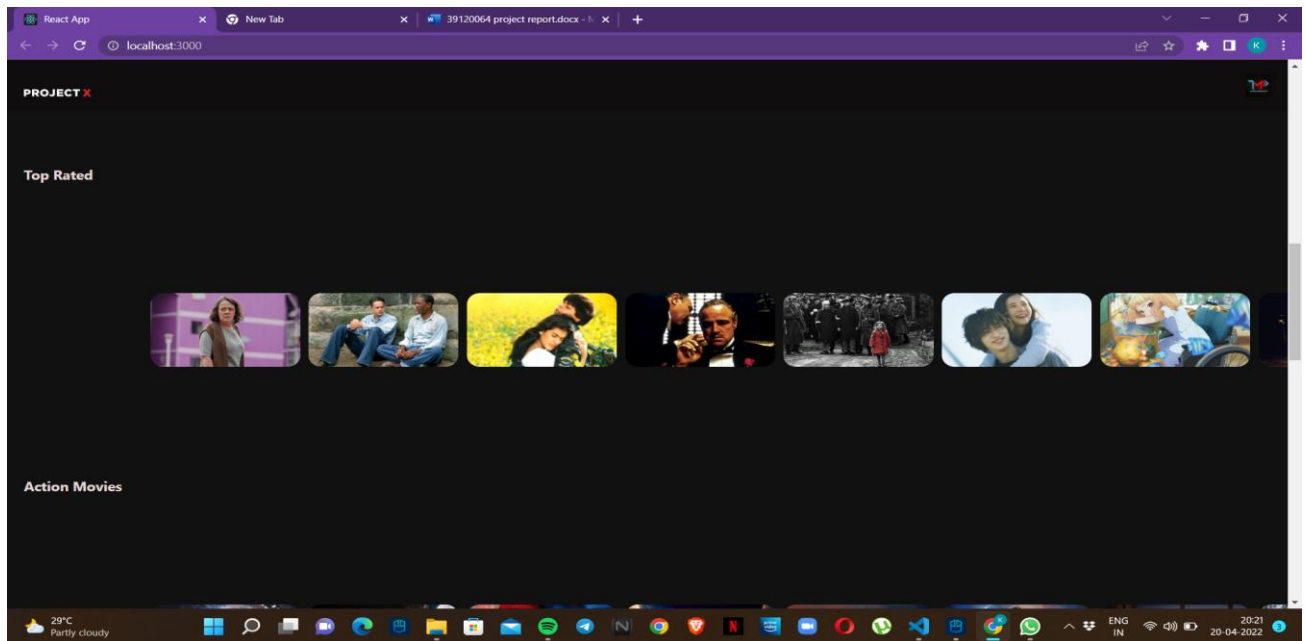


Fig 1.3: Top Rated

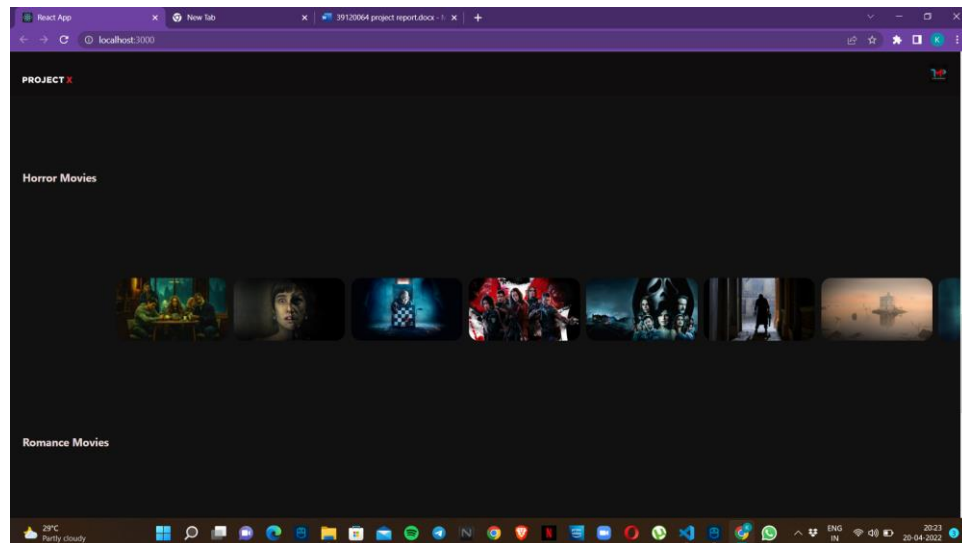


Fig 1.4: Horror Movies

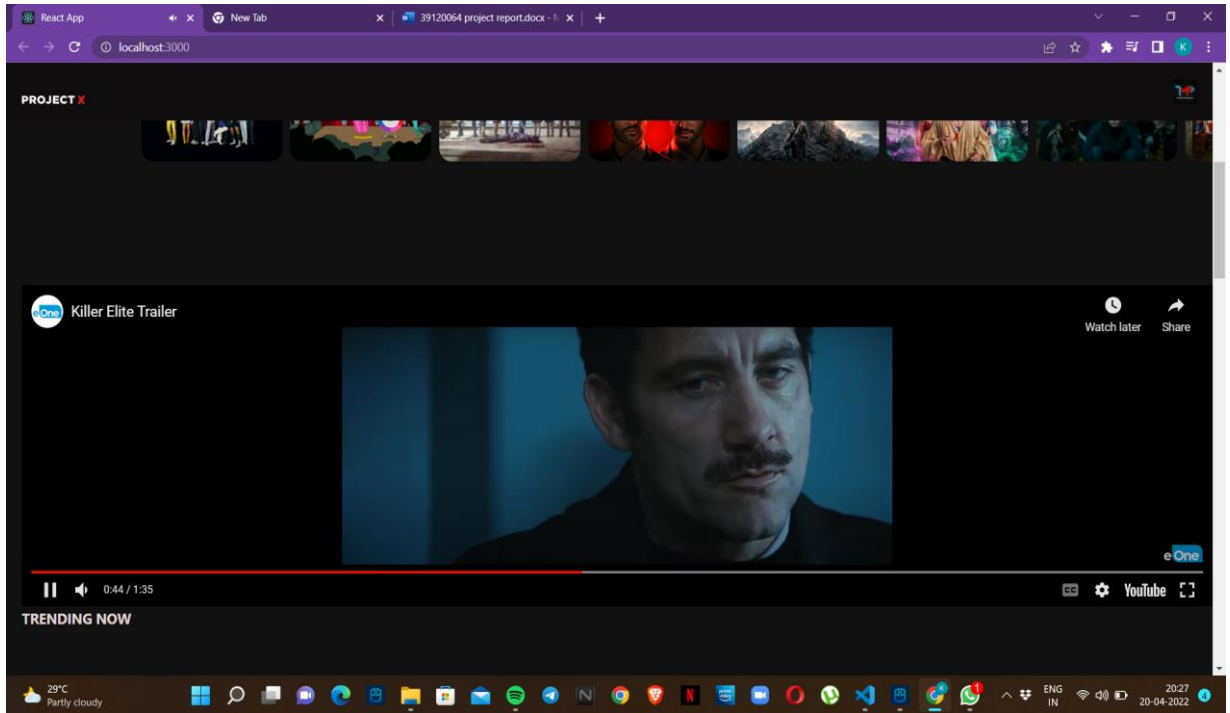
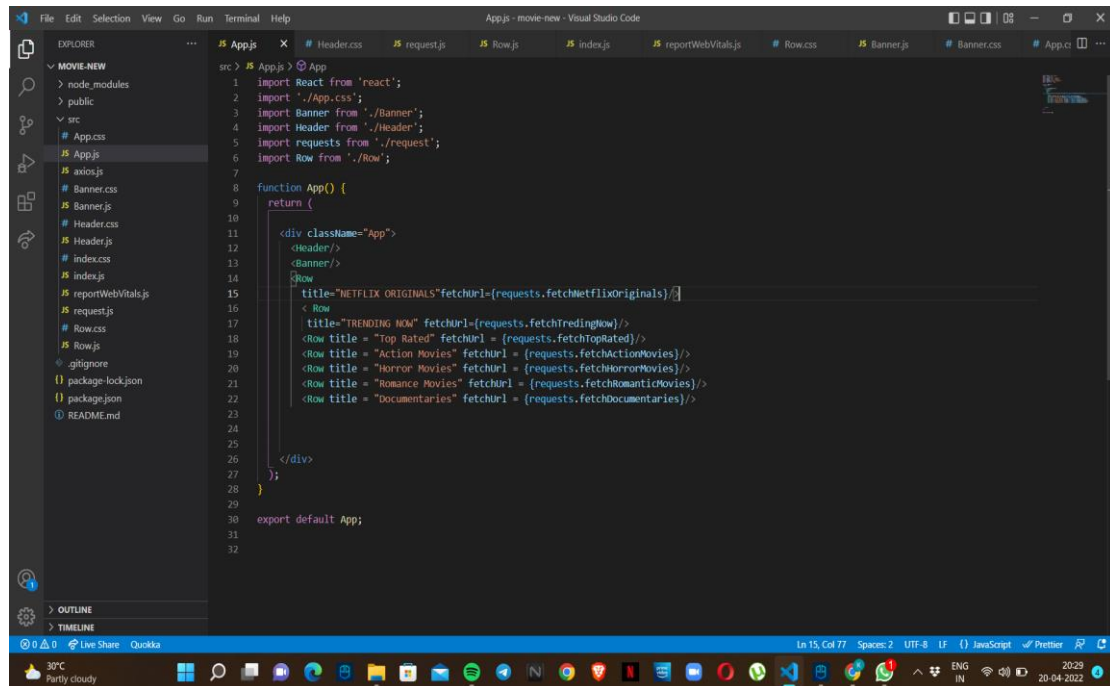


Fig 1.5 TRAILER PAGE

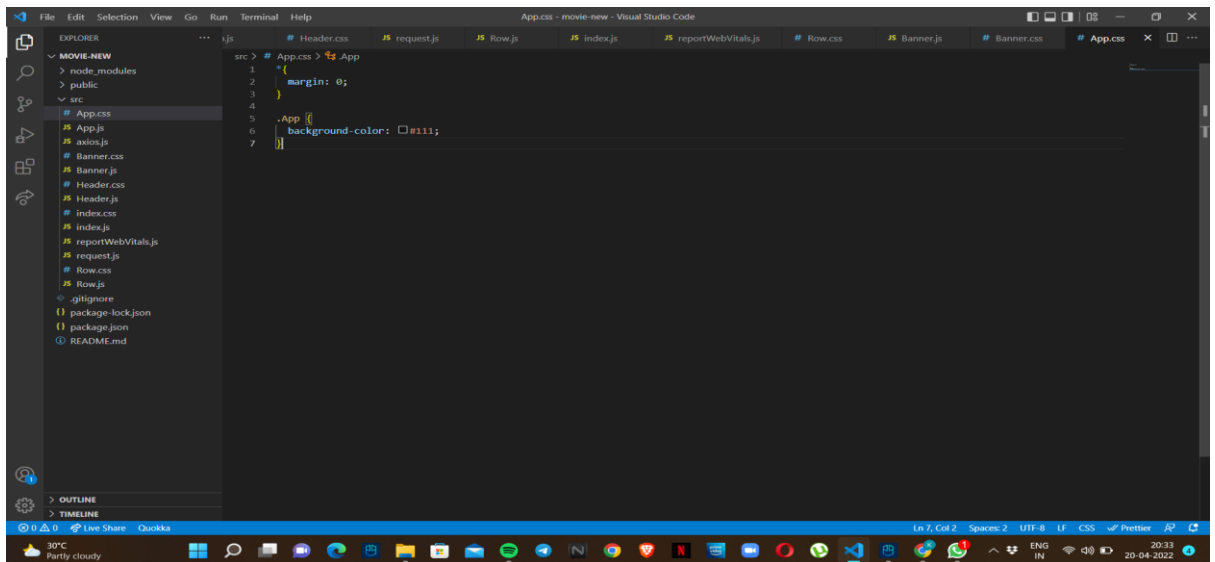
B. SOURCE CODE:

APP.JS

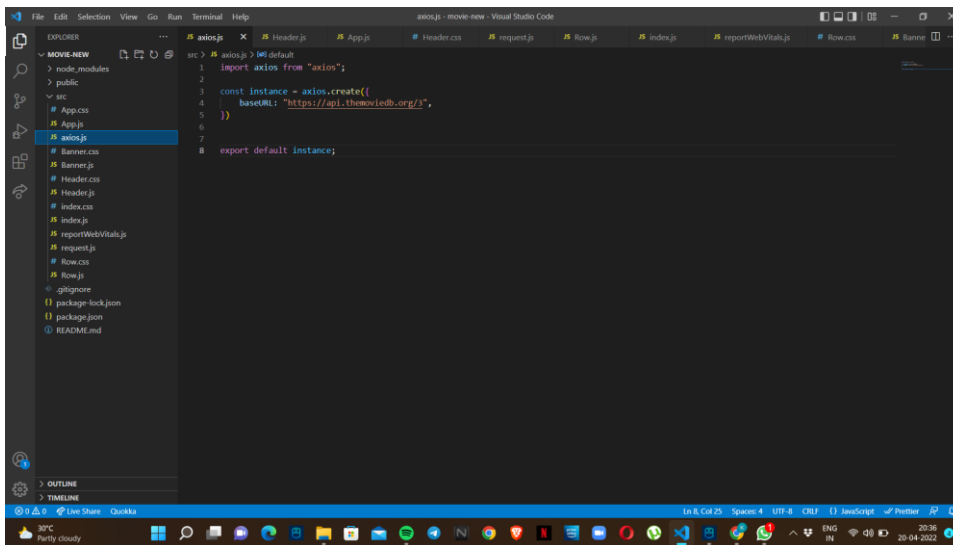


```
1 import React from 'react';
2 import './App.css';
3 import Banner from './Banner';
4 import Header from './Header';
5 import requests from './request';
6 import Row from './Row';
7
8 function App() {
9   return (
10     <div className="App">
11       <Header/>
12       <Banner/>
13       <Row
14         title="NETFLIX ORIGINALS" fetchUrl={requests.fetchNetflixOriginals}/>
15       <Row
16         title="TRENDING NOW" fetchUrl={requests.fetchTrendingNow}/>
17       <Row title = "Top Rated" fetchUrl = {requests.fetchTopRated}/>
18       <Row title = "Action Movies" fetchUrl = {requests.fetchActionMovies}/>
19       <Row title = "Horror Movies" fetchUrl = {requests.fetchHorrorMovies}/>
20       <Row title = "Romance Movies" fetchUrl = {requests.fetchRomanticMovies}/>
21       <Row title = "Documentaries" fetchUrl = {requests.fetchDocumentaries}/>
22     </div>
23   );
24 }
25
26 export default App;
```

App.CSS



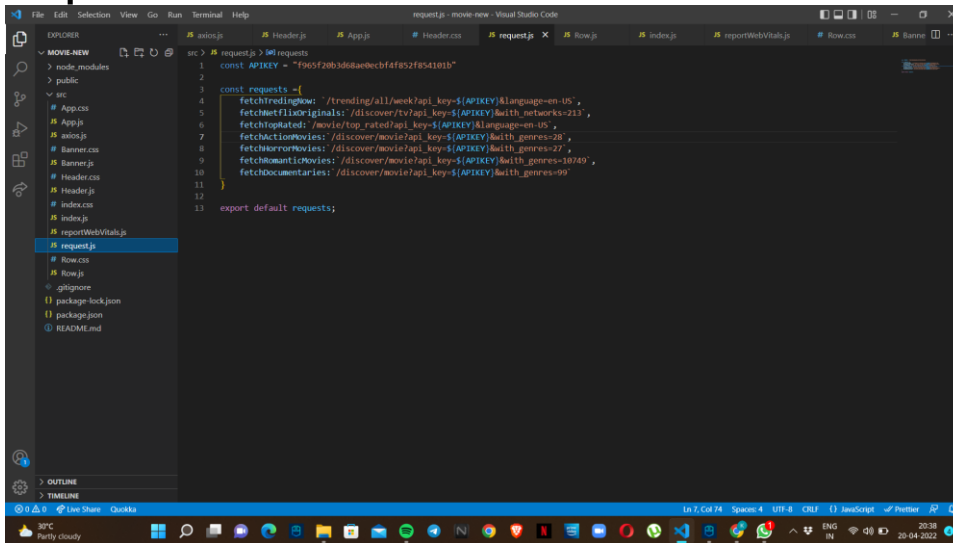
Axios.JS



The screenshot shows the Visual Studio Code interface with a project named 'movie-new'. The Explorer sidebar on the left shows a file structure with folders 'node_modules' and 'public', and a 'src' folder containing files like 'App.css', 'App.js', 'axios.js', 'Banner.css', 'Banner.js', 'Header.css', 'Header.js', 'index.css', 'index.js', 'reportWebVitals.js', 'request.js', 'Row.css', and 'Row.js'. The 'axios.js' file is selected and open in the editor. The code in 'axios.js' is as follows:

```
src > axios.js > @ default
1 import axios from "axios";
2
3 const instance = axios.create({
4   baseURL: "https://api.themoviedb.org/3",
5 })
6
7
8 export default instance;
```

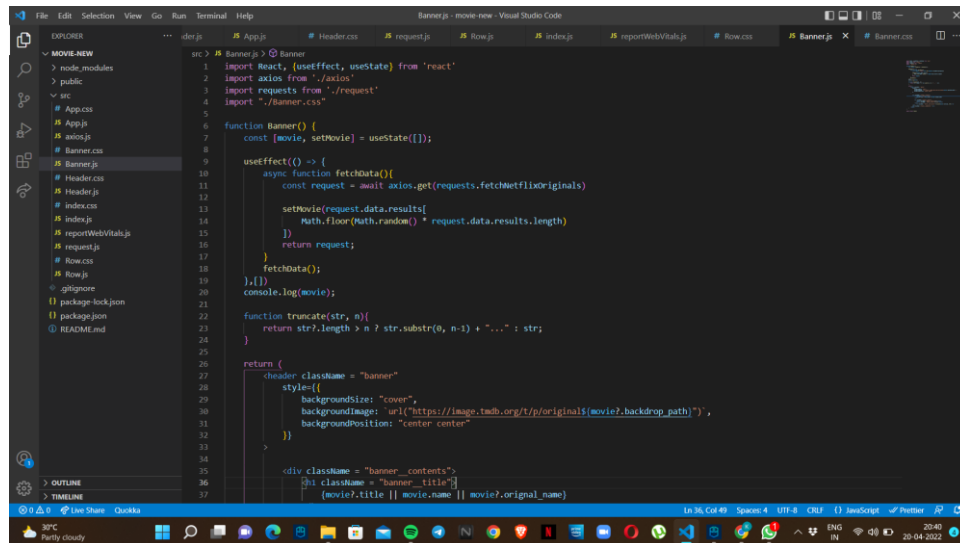
Request.JS



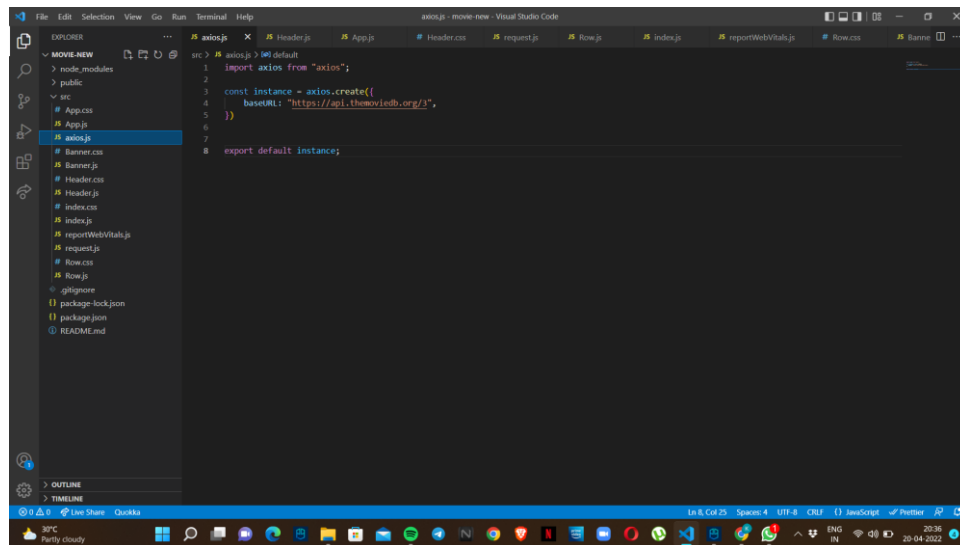
The screenshot shows the Visual Studio Code interface with a project named 'movie-new'. The Explorer sidebar on the left shows a file structure with folders 'node_modules' and 'public', and a 'src' folder containing files like 'App.css', 'App.js', 'axios.js', 'Banner.css', 'Banner.js', 'Header.css', 'Header.js', 'index.css', 'index.js', 'reportWebVitals.js', 'request.js', 'Row.css', and 'Row.js'. The 'request.js' file is selected and open in the editor. The code in 'request.js' is as follows:

```
src > request.js > @ requests
1 const APIKEY = "F9c5f2db3d68aebcf4f852f834181b"
2
3 const requests = [
4   fetchTrending: '/trending/all/week?api_key=${APIKEY}&language=en-US',
5   fetchMovieOriginals: '/discover/tv?api_key=${APIKEY}&with_networks=211',
6   fetchTopRated: '/movie/top_rated?api_key=${APIKEY}&language=en-US',
7   fetchActionMovies: '/discover/movie?api_key=${APIKEY}&with_genres=28',
8   fetchHorrorMovies: '/discover/movie?api_key=${APIKEY}&with_genres=27',
9   fetchRomanticMovies: '/discover/movie?api_key=${APIKEY}&with_genres=10749',
10  fetchDocumentaries: '/discover/movie?api_key=${APIKEY}&with_genres=99'
11 ]
12
13 export default requests;
```

Banner.JS

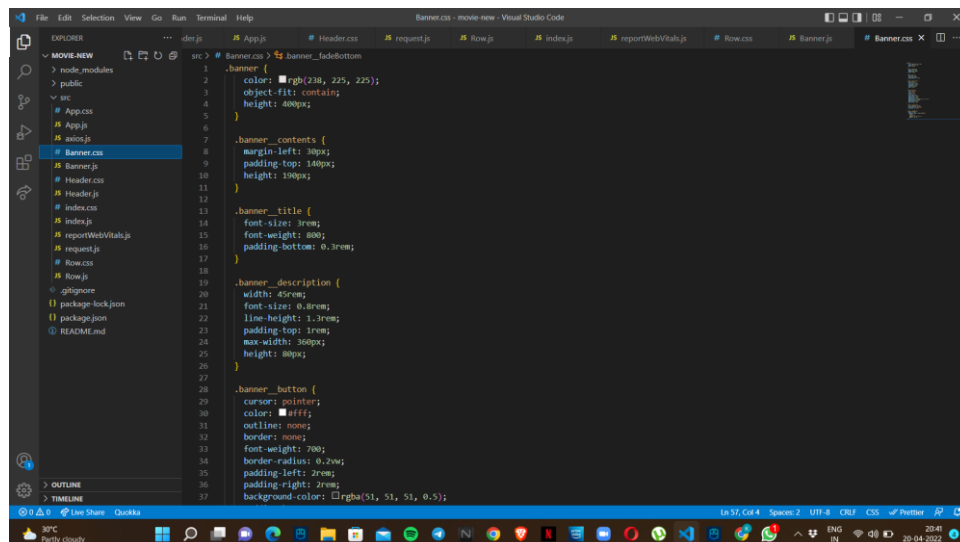


```
src > # Banner.js > @ Banner
1 import React, {useEffect, useState} from 'react'
2 import axios from './axios'
3 import requests from './request'
4 import './Banner.css'
5
6 function Banner() {
7   const [movie, setMovie] = useState([]);
8
9   useEffect(() => {
10     async function fetchData() {
11       const request = await axios.get(requests.fetchNetflixOriginals)
12       setMovie(request.data.results[
13         Math.floor(Math.random() * request.data.results.length)
14       ])
15       return request;
16     }
17     fetchData();
18     console.log(movie);
19   }, [])
20
21   function truncate(str, n) {
22     return str.length > n ? str.substr(0, n-1) + "...": str;
23   }
24
25   return (
26     <div className="banner">
27       <div>
28         <img alt="Movie backdrop image" data-bbox="102 135 705 376"/>
29         <div>
30           <div>
31             <div>
32               <div>
33                 <div>
34                   <div>
35                     <div>
36                       <div>
37                         <div>
```

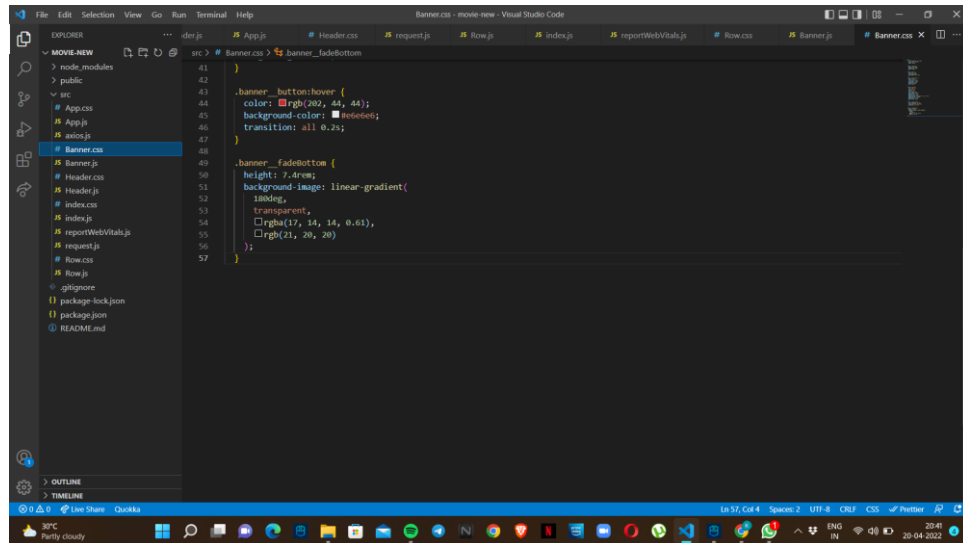


```
src > # axios.js > @ default
1 import axios from 'axios';
2
3 const instance = axios.create({
4   baseURL: "https://api.themoviedb.org/3",
5 })
6
7 export default instance;
```

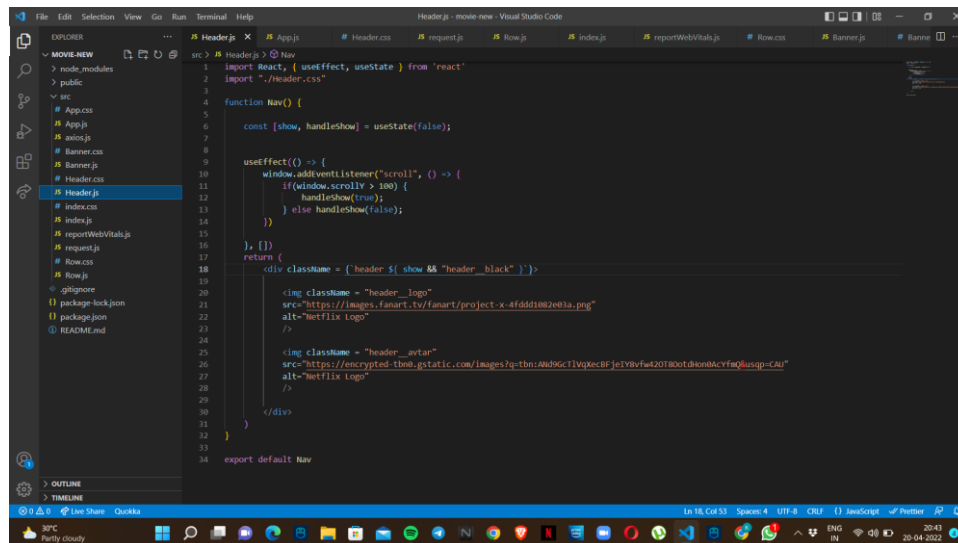
Banner.CSS



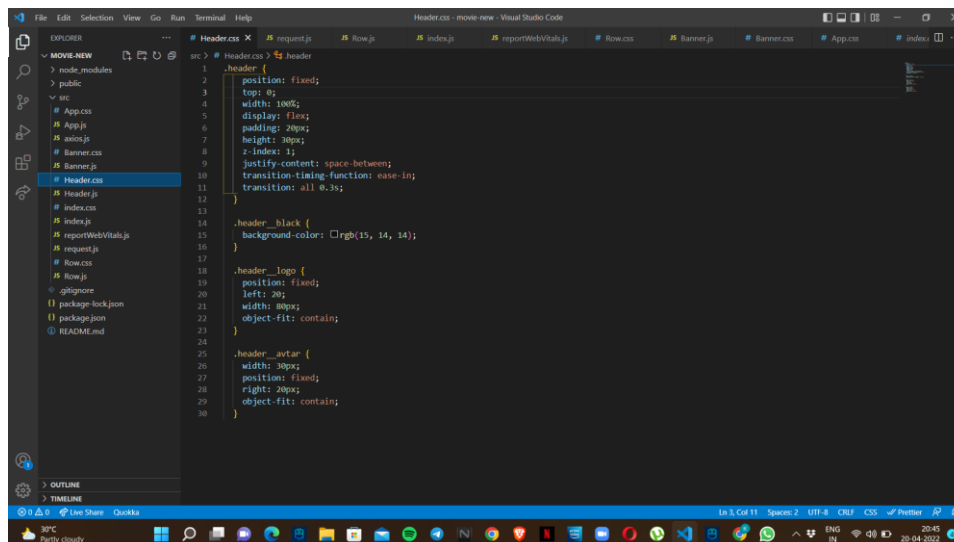
```
src > # Banner.css > @ banner_fadeBottom
1 .banner {
2   color: #fff;
3   object-fit: contain;
4   height: 400px;
5 }
6
7 .banner_contents {
8   margin-left: 30px;
9   padding-top: 140px;
10  height: 100px;
11 }
12
13 .banner_title {
14   font-size: 3rem;
15   font-weight: 800;
16   padding-bottom: 0.3rem;
17 }
18
19 .banner_description {
20   width: 45rem;
21   font-size: 0.8rem;
22   line-height: 1.3rem;
23   padding-top: 1rem;
24   max-width: 300px;
25   height: 80px;
26 }
27
28 .banner_button {
29   cursor: pointer;
30   color: #fff;
31   outline: none;
32   border: none;
33   font-weight: 700;
34   border-radius: 0.2rem;
35   padding-left: 2rem;
36   padding-right: 2rem;
37   background-color: linear-gradient(to right, #f0f0f0, #f0f0f0);
```



Header.JS



Header.css



Row.JS

```
File Edit Selection View Go Run Terminal Help
Row.js - movie - Visual Studio Code

EXPLORE
  > MOVIE-NEW
  > node_modules
  > public
  > src
    # App.css
    # App.js
    # axios.js
    # Banner.css
    # Banner.js
    # Header.css
    # Header.js
    # index.js
    # reportWebVitals.js
    # request.js
    # Row.css
    # Row.js
    # ignore
  () package-lock.json
  () package.json
  () README.md

OUTLINE
TIMELINE

src > Row.js
1  import React, { useEffect, useState } from 'react'
2  import './Row.css';
3  import axios from './axios';
4  import Youtube from 'react-youtube';
5  import movieTrailer from 'movie-trailer';
6
7  const baseUrl = "https://image.tmdb.org/t/p/original";
8
9  function Row({title, fetchUrl, isLargeRow}) {
10   const [movies, setMovies] = useState([]);
11   const [trailerUrl, setTrailerUrl] = useState("");
12   useEffect(() => {
13     async function fetchData() {
14       const request = await axios.get(fetchUrl);
15       setMovies(request.data.results);
16       return request;
17     }
18     fetchData();
19   }, [fetchUrl]);
20
21   const opts = {
22     height: "390px",
23     width: "100%",
24     playerVars: {
25       autoplay: 1,
26     },
27   };
28
29   const handleClick = (movie) => {
30     if (trailerUrl) {
31       setTrailerUrl("");
32     } else {
33       movieTrailer(movie.name || "")
34         .then(url => {
35           const urlParams = new URLSearchParams(new URL(url).search);
36           setTrailerUrl(urlParams.get("v"));
37         });
38     }
39   };
40
41   .catch(error => console.log(error));
42 }
43
44 return (
45   <div className = "row">
46     <div className = "row_posters">
47       <div className = "row_poster">
48         <img
49           key={movie.id}
50           onClick={() => handleClick(movie)}
51           className = {"row_poster ${isLargeRow ? "row_poster_large" : ""}"}
52           src = {`${baseUrl}${isLargeRow ? movie.poster_path : movie.backdrop_path}`}
53           alt = {movie.name} />
54         </div>
55       </div>
56       <div>
57         <video
58           id={trailerUrl}
59           <YouTube videoId={trailerUrl} opts={opts} />
60       </div>
61     </div>
62   </div>
63 );
64
65 export default Row;
```

```
File Edit Selection View Go Run Terminal Help
Row.js - movie - Visual Studio Code

EXPLORE
  > MOVIE-NEW
  > node_modules
  > public
  > src
    # App.css
    # App.js
    # axios.js
    # Banner.css
    # Banner.js
    # Header.css
    # Header.js
    # index.js
    # reportWebVitals.js
    # request.js
    # Row.css
    # Row.js
    # ignore
  () package-lock.json
  () package.json
  () README.md

OUTLINE
TIMELINE

src > Row.js
1  import React, { useEffect, useState } from 'react'
2  import './Row.css';
3  import axios from './axios';
4  import Youtube from 'react-youtube';
5  import movieTrailer from 'movie-trailer';
6
7  const baseUrl = "https://image.tmdb.org/t/p/original";
8
9  function Row({title, fetchUrl, isLargeRow}) {
10   const [movies, setMovies] = useState([]);
11   const [trailerUrl, setTrailerUrl] = useState("");
12   useEffect(() => {
13     async function fetchData() {
14       const request = await axios.get(fetchUrl);
15       setMovies(request.data.results);
16       return request;
17     }
18     fetchData();
19   }, [fetchUrl]);
20
21   const opts = {
22     height: "390px",
23     width: "100%",
24     playerVars: {
25       autoplay: 1,
26     },
27   };
28
29   const handleClick = (movie) => {
30     if (trailerUrl) {
31       setTrailerUrl("");
32     } else {
33       movieTrailer(movie.name || "")
34         .then(url => {
35           const urlParams = new URLSearchParams(new URL(url).search);
36           setTrailerUrl(urlParams.get("v"));
37         });
38     }
39   };
40
41   .catch(error => console.log(error));
42 }
43
44 return (
45   <div className = "row">
46     <div className = "row_posters">
47       <div className = "row_poster">
48         <img
49           key={movie.id}
50           onClick={() => handleClick(movie)}
51           className = {"row_poster ${isLargeRow ? "row_poster_large" : ""}"}
52           src = {`${baseUrl}${isLargeRow ? movie.poster_path : movie.backdrop_path}`}
53           alt = {movie.name} />
54         </div>
55       </div>
56       <div>
57         <video
58           id={trailerUrl}
59           <YouTube videoId={trailerUrl} opts={opts} />
60       </div>
61     </div>
62   </div>
63 );
64
65 export default Row;
```

Row.CSS

```
1  .row {
2    margin-left: 20px;
3    font-weight: bold;
4    font-size: 1.08rem;
5    color: rgb(227, 214, 214);
6  }
7  .row_posters {
8    display: flex;
9    overflow: hidden;
10   overflow-x: scroll;
11   padding: 150px;
12 }
13
14 .row_posters::-webkit-scrollbar {
15   display: none;
16 }
17
18 .row_poster {
19   object-fit: contain;
20   width: 100%;
21   max-height: 100px;
22   margin-right: 10px;
23   transition: transform 450ms;
24   border-radius: 1w;
25 }
26
27 .row_poster:hover {
28   transform: scale(1.08);
29 }
30
31 .row_posterIarge {
32   max-height: 250px;
33 }
34
35 .row_posterIarge:hover {
36   transform: scale(1.09);
37 }
```