

Responsive Movie Application

PROJECT REPORT

For
MINOR PROJECT

BACHELOR OF TECHNOLOGY

Computer Science & Engineering

SUBMITTED BY

Deepak	Aman Kumar Jha
196301031	196301005

UNDER THE SUPERVISION OF

Dr. Nishant Kumar

Assistant Professor,



**Faculty of Engineering and Technology
Gurukula Kangri (Deemed to be University)
Haridwar**

December 2022

**Report for Minor Project
in**

Computer Science & Engineering

Proposed Topic : Responsive Movie Application

(Signature of Supervisor)

Dr. Nishant Kumar

Assistant Professor,

Department of Computer Science &
Engineering, Faculty of Engineering and
Technology,

Gurukula Kangri (Deemed to be University),
Haridwar.

(Signature of Students)

Deepak
196301031

Aman Kumar Jha
196301005

Forwarded by:

Head of the Department
Department of Computer Science & Engineering
Faculty of Engineering and Technology
Gurukula Kangri (Deemed to be University), Haridwar

Introduction:

This project aims to introduce a movie application that provides information to users about latest movies , top rated movies , top rated tv shows and popular tv shows.

- This project aims to introduce a movie application that provides information to users about latest movies , top rated movies , top rated tv shows and popular tv shows.
- User can search any movie. **User can search any movies and tv shows and get all information about them.**
- Users will get a better user interface and sleek design. **User will experience better user interface and sleek design.**

Features of the app and website

User can search movies and tv shows

User can get all information about a movie or tv shows like its cast, total revenue and its all official trailer and much more

User can also get recommendation movies and animes.

Application gives same experience as Netflix , amazon prime video

Developed Features:

1. Web frontend
 - a. Movie Home Page
 - b. Single Movie page
 - c. Movie Search Page

Languages and Frameworks Used:

1. Frontend Development
 - a. Website
 - i. React JS
 - ii. HTML
 - iii. CSS
 - iv. JavaScript

Development Process:

1. Set up the project environment: - Install and configure necessary software such as Node.js, React.js, and TMDB API. - Create project directories and files.
2. Develop the website: - Create user interface components using React.js. - Create routes and logic for the user interface. - Test the website.
3. Responsive Website using vanilla CSS , media queries , and semantic html
4. Using third party API fetching movie data from database of TMDB API.

Screenshots:

The screenshot shows a code editor interface with the following details:

- File Path:** src/App.js
- Content:** The code defines the main application component (App) which renders a div with the class 'app'. It uses a BrowserRouter to handle routes. The routes include paths for '/', '/movies', '/tv', and '/:type/:movieId'. It also includes imports for Body, Movie, TV, and SingleMovie components from ./pages.

```
// const Body = React.lazy(() => import('./pages/Body'))
// const Movie = React.lazy(() => import('./nav-pages/Movie'))
// const TV = React.lazy(() => import('./nav-pages/TV'))
// const SingleMovie = React.lazy(() => import('./pages/SingleMovie'))

function App() {
  return (
    <div className='app'>
      <BrowserRouter>
        <Nav />

        <Routes>
          <Route path="/" element={<Body />} />
          <Route path="/movies" element={<Movie />} />
          <Route path="/tv" element={<TV />} />

          <Route path="/:type/:movieId" element={<SingleMovie />} />
        </Routes>

        <Footer />
      </BrowserRouter>
    </div>
  );
}

export default App;
```

- Explorer View:** Shows the project structure with folders like build, node_modules, public, and src, along with files like App.css, index.css, index.js, and Test.jsx.
- Bottom Status Bar:** Displays information such as 'Ln 1, Col 1', 'Spaces: 2', 'UTF-8', 'LF', 'JavaScript', 'Go Live', 'Prettier', and system icons for search, file operations, and network.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "NETFLIX-CLONE" with files like App.js, App.css, index.css, and Test.jsx.
- Code Editor:** Displays the content of App.js, which defines a functional component "App" that sets up a Router with various routes including "/" (Body), "/movies" (Movie), "/tv" (TV), and a dynamic route ":type/:movieId" (SingleMovie). It also includes a Footer component.
- Bottom Bar:** Includes icons for search, file operations, and extensions, along with status information: Ln 1, Col 1, Spaces: 2, UTF-8, LF, JavaScript, Go Live, Prettier, and a date/time stamp: 08-12-2022.

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "NETFLIX-CLONE" with files like App.js, Body.js, and Row.jsx.
- Code Editor:** Displays the content of Row.jsx, which is a functional component that fetches movie data from an API and displays it in a row. It uses useState and useEffect hooks, along with axios for API requests.
- Bottom Bar:** Includes icons for search, file operations, and extensions, along with status information: Ln 1, Col 1, Spaces: 2, UTF-8, CRLF, JavaScript React, Go Live, Prettier, and a date/time stamp: 08-12-2022.

```
src > nav-pages > Movie.jsx ...
1 import React, { useState, useEffect } from "react";
2 import axios from "../request/axios";
3 import { Link } from "react-router-dom";
4 import "./Movies.css";
5
6 import TvBanner from "../components/TvBanner";
7
8 const Movie = () => {
9   const [isLoading, setIsLoading] = useState(false);
10  const [movies, setMovies] = useState([]);
11  const [searchMovie, setSearchMovie] = useState("");
12
13  const [resultMovie, setResultMovie] = useState([]);
14
15  const APIKEY = "8a2705651c885ab1d285fc80ee021c5";
16  const baseUrl = "https://image.tmdb.org/t/p/original";
17  const apiUrl = `https://api.themoviedb.org/3/search/movie?api_key=${APIKEY}&language=en-US&page=1&include_adult=false&query=${searchMovie}`;
18  const movieUrl = `https://api.themoviedb.org/3/movie/popular?api_key=${APIKEY}&language=en-US`;
19
20  useEffect(() => {
21    const fetchData = async () => {
22      const request = await axios.get(movieUrl);
23      // console.log(request);
24      setMovies(request.data.results);
25    };
26    fetchData();
27  }, [movieUrl]);
28
29  const handleSearch = async () => {
30    setIsLoading(true);
31
32    const res = await fetch(apiUrl);
33    const data = await res.json();
34
35    setResultMovie(data.results);
36    setIsLoading(false);
37  };
38
39  if (isLoading) {
40    return <h2>Loading...</h2>;
41  }
42  if (resultMovie.length > 0) {
43    return (
44      <>
45        <TvBanner title="Movies" />
46        <div className="input-container">
47          <input type="text" value={searchMovie} onChange={(e) => setSearchMovie(e.target.value)} />
48        </div>
49        <div className="list">
50          {resultMovie.map((movie) => (
51            <div>
52              <img alt={movie.poster_path} />
53              {movie.title}
54            </div>
55          ))}
56        </div>
57      </>
58    );
59  }
60
61  return <h2>No results found!</h2>;
62}
63
64
```

Ln 1, Col 1 | Spaces: 2 | UTF-8 | CRLF | JavaScript React | Go Live | Prettier

08-12-2022

```
src > nav-pages > TV.jsx ...
1 import React, { useState, useEffect } from "react";
2 import axios from "../request/axios";
3 import { Link } from "react-router-dom";
4 import "./TV.css";
5
6 import ScrollToTop from "react-scroll-to-top";
7 import LazyLoadImage from "react-lazy-load-image-component";
8 import "react-lazy-load-image-component/src/effects/blur.css";
9
10 import TvBanner from "../components/TvBanner";
11
12 const TV = () => {
13  const [isLoading, setIsLoading] = useState(false);
14  const [movies, setMovies] = useState([]);
15  const [searchMovie, setSearchMovie] = useState("");
16
17  const [resultMovie, setResultMovie] = useState([]);
18
19  const APIKEY = "8a2705651c885ab1d285fc80ee021c5";
20  const baseUrl = "https://image.tmdb.org/t/p/original";
21  const apiUrl = `https://api.themoviedb.org/3/search/tv?api_key=${APIKEY}&language=en-US&page=1&include_adult=false&query=${searchMovie}`;
22  const movieUrl = `https://api.themoviedb.org/3/tv/popular?api_key=${APIKEY}&language=en-US`;
23
24  useEffect(() => {
25    setIsLoading(true);
26    const fetchData = async () => {
27      const request = await axios.get(movieUrl);
28      // console.log(request);
29      setMovies(request.data.results);
30      setResultMovie(request.data.results);
31      setIsLoading(false);
32    };
33    fetchData();
34  }, [movieUrl]);
35
36  const handleSearch = async () => {
37    setIsLoading(true);
38
39    const res = await fetch(apiUrl);
40    const data = await res.json();
41
42    setResultMovie(data.results);
43    setIsLoading(false);
44  };
45
46  if (isLoading) {
47    return <h2>Loading...</h2>;
48  }
49
50  return (
51    <>
52      <TvBanner title="TV Shows" />
53      <div className="input-container">
54        <input type="text" value={searchMovie} onChange={(e) => setSearchMovie(e.target.value)} />
55      </div>
56      <div className="list">
57        {resultMovie.map((movie) => (
58          <div>
59            <img alt={movie.poster_path} />
60            {movie.title}
61          </div>
62        ))}
63      </div>
64    </>
65  );
66}
67
68
```

Ln 1, Col 1 | Spaces: 2 | UTF-8 | CRLF | JavaScript React | Go Live | Prettier

08-12-2022

S Easy to use Online PDF editor × React App × +

localhost:3001

NETFLIX

NOS Studio WK 22

Play My List

Movies TV Series

NOS Studio WK

Trending Movies

HEX JUMP. PULL. PRAY. DIE. KAYLA ADAMS MATTHEW HOLCOMB

ONE WAY

Windows taskbar icons: File Explorer, Search, Task View, Microsoft Edge, WhatsApp, and others.

ENG IN 08:33 08-12-2022

THANK YOU