FULL STACK DEVELOPMENT INTERNSHIP PROJECT REPORT

Project Title:

Free Movie Generator-A Full Stack Internship Project

Submitted by:

Aina Sudeep

B.Tech 1st Year, CSE Department

FISAT

Submitted to:

Pacelab

Kadavanthra

ABSTRACT

This project is a full-stack web application developed using react for the frontend, tailwind css for styling, and express.js for the backend. The main goal was to fetch real-time movie data from the omdb api and present it in a structured, responsive layout. The express backend acts as a bridge between the public api and the react frontend, providing filtered data to improve performance and handle api requests securely. The frontend dynamically displays the movie data in a card format, showing each movie's title, release year, and poster image, along with a link to watch the movie trailer. This project demonstrates the core workflow of full-stack development and highlights how different technologies can be combined to create a functional and visually appealing movie search application.

INTRODUCTION

The rise of web-based applications has significantly increased the need for developers to understand how data flows between the backend and the frontend. This project was developed as a practical exercise to learn how to build a basic full-stack web application by integrating a public movie API. The report outlines the journey of building a functional web app that fetches and displays real-time movie data using a React frontend, an Express backend, and Tailwind CSS for styling

PROJECT SUMMARY

The task was to find an open movie API, set up a backend server using Express to fetch data from it, and then use a React frontend to display the data using simple styling with Tailwind CSS. This project is a full-stack web application that presents a collection of movies by integrating a public API with a React frontend and an Express.js backend. The backend, built using index.js, acts as a middleware to securely fetch movie data from the OMDb API and serve it to the frontend. The React frontend consists of App.jsx, which handles the search functionality and API calls, and Card.jsx, which is responsible for displaying individual movie details such as the poster image, title, and release year in a clean card layout. Tailwind CSS is used throughout to create a responsive and visually appealing user interface. Each card also includes a link to watch the movie's trailer on YouTube. This project demonstrates a practical understanding of API consumption, component-based frontend architecture, and full-stack integration using modern JavaScript frameworks

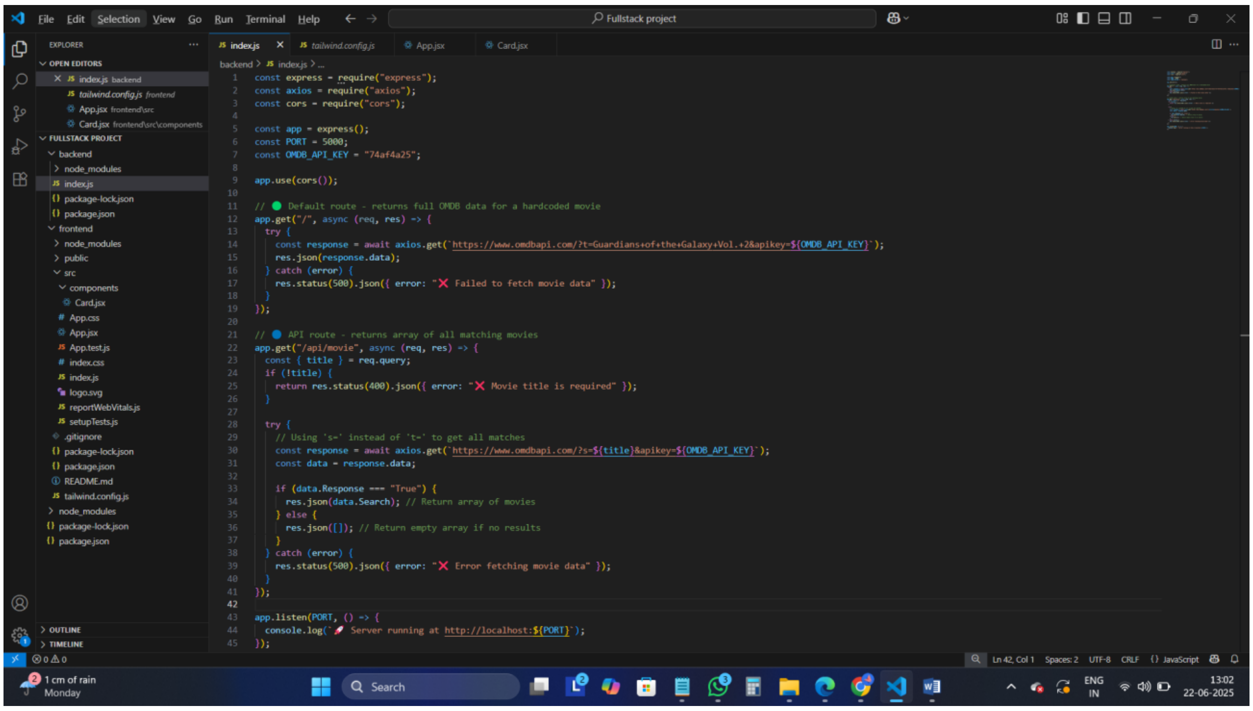
FEATURES

- Public API Integration – Connects to the OMDb API to fetch real-time movie data based on user search input.
- Express Backend Server – Creates a /api/movie endpoint to serve filtered movie data securely to the frontend.
- Filtered API Response – Processes and filters movie search results to display only relevant data for a cleaner interface.
- React-Powered Frontend – Uses React components to dynamically render movie cards based on the fetched data.
- Responsive Layout – Tailwind CSS enables a clean, mobile-friendly design that adjusts to various screen sizes.
- Movie Cards – Each card displays the movie's poster, title, release year, and a trailer link.
- Fallbacks for Missing Data – Provides placeholder images or "N/A" text when poster images or details are unavailable.
- Modular Code Structure – Organized into reusable components such as App.jsx and Card.jsx for better code maintenance.
- Error Handling – Manages API failures and empty search results gracefully, providing user-friendly messages.
- Clean UI Design – Minimalistic interface with easy navigation and clear visual presentation of movie data.

IMPLEMENTATION

1.Backend

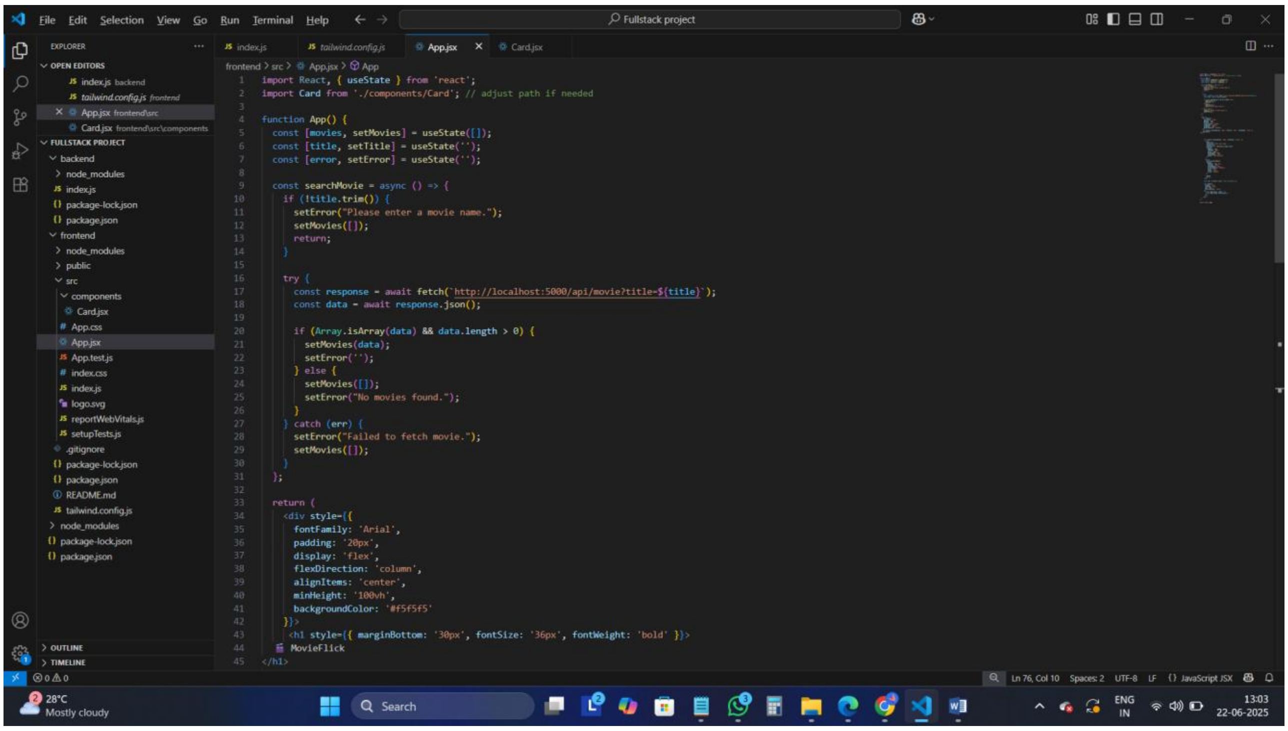
Index.js:



```javascript
const express = require("express");
const axios = require("axios");
const cors = require("cors");

const app = express();
const PORT = 5000;
const OMDB_API_KEY = "74af4a25";

app.use(cors());

// 🟢 Default route - returns full OMDB data for a hardcoded movie
app.get("/", async (req, res) => {
  try {
    const response = await axios.get(`https://www.omdbapi.com/?t=Guardians+of+the+Galaxy+Vol.+2&apikey=${OMDB_API_KEY}`);
    res.json(response.data);
  } catch (error) {
    res.status(500).json({ error: "❌ Failed to fetch movie data" });
  }
});

// 🔵 API route - returns array of all matching movies
app.get("/api/movie", async (req, res) => {
  const { title } = req.query;
  if (!title) {
    return res.status(400).json({ error: "❌ Movie title is required" });
  }

  try {
    // Using 's=' instead of 't=' to get all matches
    const response = await axios.get(`https://www.omdbapi.com/?s=${title}&apikey=${OMDB_API_KEY}`);
    const data = response.data;

    if (data.Response === "True") {
      res.json(data.Search); // Return array of movies
    } else {
      res.json([]); // Return empty array if no results
    }
  } catch (error) {
    res.status(500).json({ error: "❌ Error fetching movie data" });
  }
});

app.listen(PORT, () => {
  console.log(`🚀 Server running at http://localhost:${PORT}`);
});
```

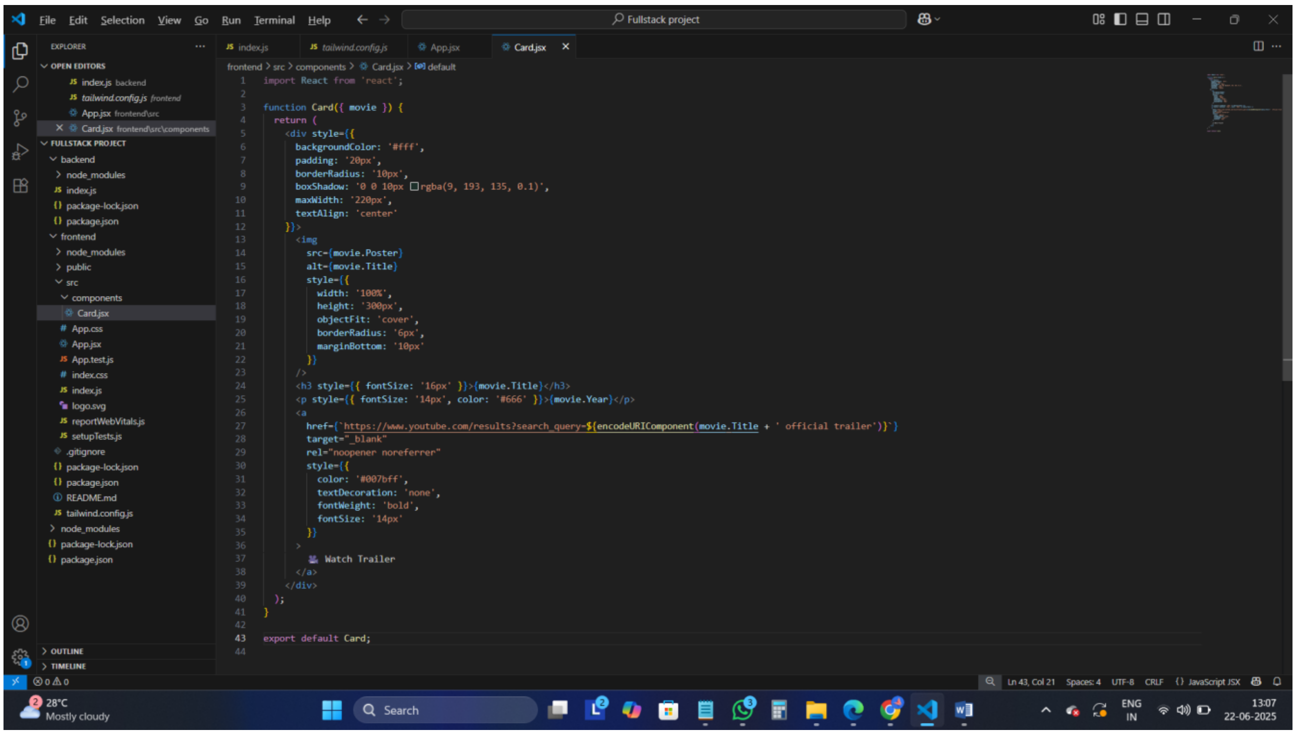
2.Frontend

App.jsx:



```jsx
import React, { useState } from 'react';
import Card from './components/Card'; // adjust path if needed

function App() {
  const [movies, setMovies] = useState([]);
  const [title, setTitle] = useState('');
  const [error, setError] = useState('');

  const searchMovie = async () => {
    if (!title.trim()) {
      setError("Please enter a movie name.");
      setMovies([]);
      return;
    }

    try {
      const response = await fetch(`http://localhost:5000/api/movie?title=${title}`);
      const data = await response.json();

      if (Array.isArray(data) && data.length > 0) {
        setMovies(data);
        setError('');
      } else {
        setMovies([]);
        setError("No movies found.");
      }
    } catch (err) {
      setError("Failed to fetch movie.");
      setMovies([]);
    }
  };

  return (
    <div style={{
      fontFamily: 'Arial',
      padding: '20px',
      display: 'flex',
      flexDirection: 'column',
      alignItems: 'center',
      minHeight: '100vh',
      backgroundColor: '#f5f5f5'
    }}>
     <h1 style={{ marginBottom: '30px', fontSize: '36px', fontWeight: 'bold' }}>
   🎬 MovieFlick
</h1>
```

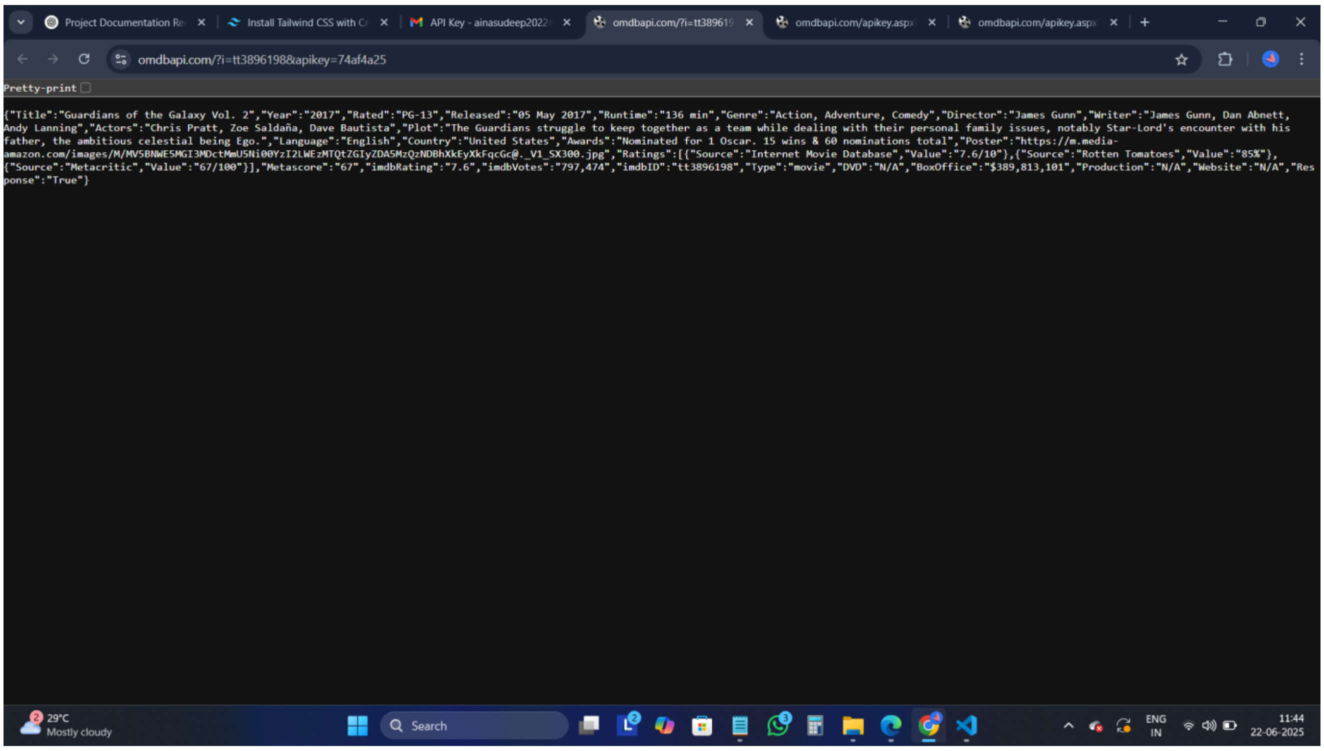
Card.jsx:



```
import React from 'react';

function Card({ movie }) {
  return (
    <div style={{
      backgroundColor: '#fff',
      padding: '20px',
      borderRadius: '10px',
      boxShadow: '0 0 10px rgba(9, 193, 135, 0.1)',
      maxWidth: '220px',
      textAlign: 'center'
    }}>
      <img
        src={movie.Poster}
        alt={movie.Title}
        style={{
          width: '100%',
          height: '300px',
          objectFit: 'cover',
          borderRadius: '6px',
          marginBottom: '10px'
        }}
      />
      <h3 style={{ fontSize: '16px' }}>{movie.Title}</h3>
      <p style={{ fontSize: '14px', color: '#666' }}>{movie.Year}</p>
      <a
        href={`https://www.youtube.com/results?search_query=${encodeURIComponent(movie.Title + ' official trailer')}`}
        target="_blank"
        rel="noopener noreferrer"
        style={{
          color: '#007bff',
          textDecoration: 'none',
          fontWeight: 'bold',
          fontSize: '14px'
        }}
      >
        🎥 Watch Trailer
      </a>
    </div>
  );
}

export default Card;

```

Movies API

Server started on http://localhost:5000 successfully

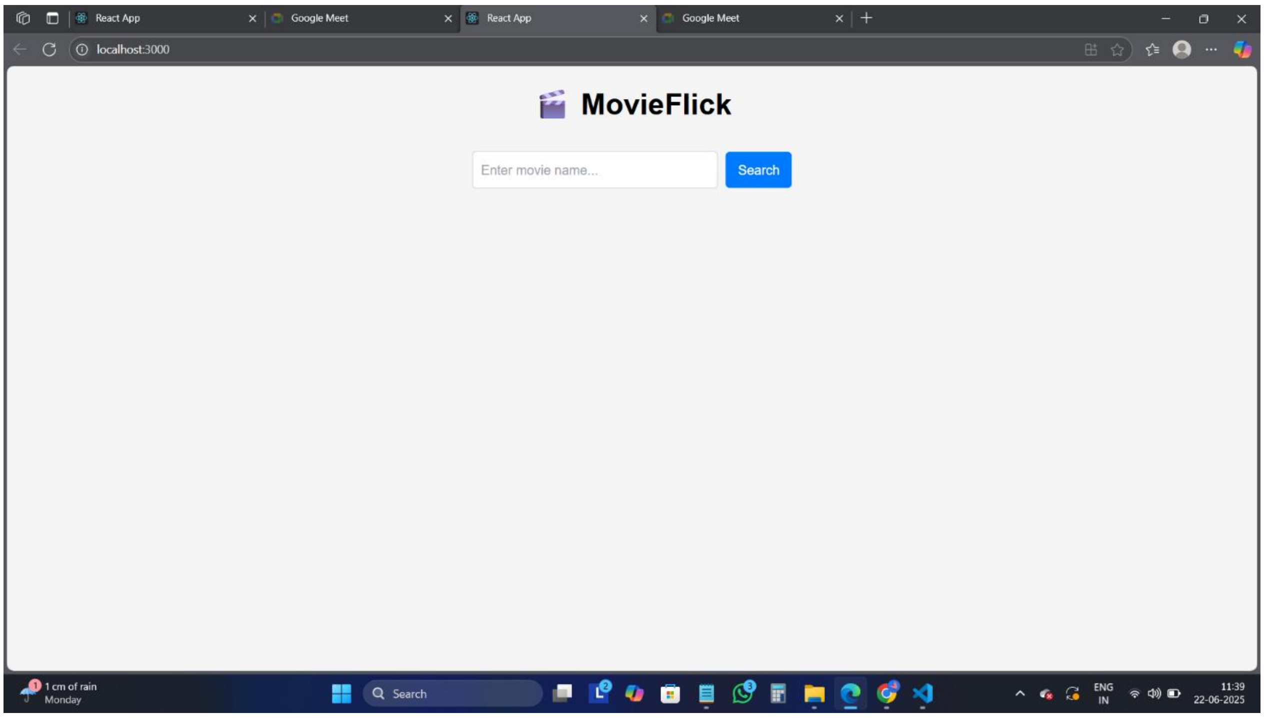


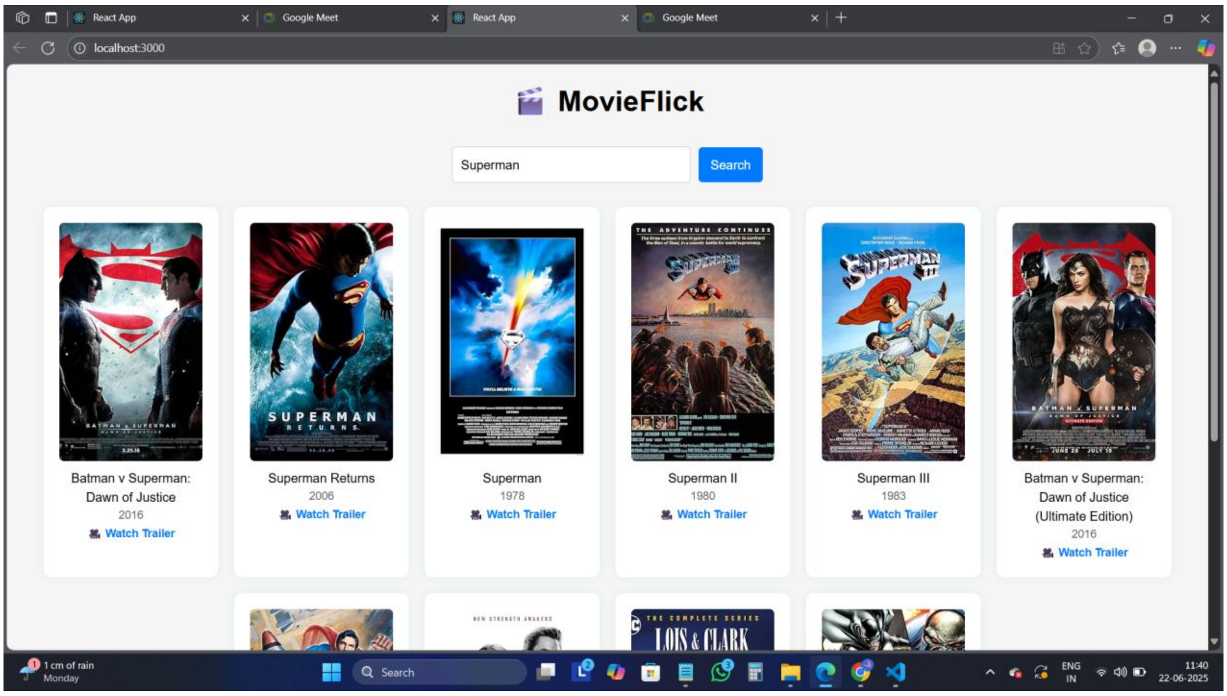
omdbapi.com/?i=tt3896198&apikey=74af4a25

```
{"Title":"Guardians of the Galaxy Vol. 2","Year":"2017","Rated":"PG-13","Released":"05 May 2017","Runtime":"136 min","Genre":"Action, Adventure, Comedy","Director":"James Gunn","Writer":"James Gunn, Dan Abnett,
Andy Lanning","Actors":"Chris Pratt, Zoe Saldaña, Dave Bautista","Plot":"The Guardians struggle to keep together as a team while dealing with their personal family issues, notably Star-Lord's encounter with his
father, the ambitious celestial being Ego.","Language":"English","Country":"United States","Awards":"Nominated for 1 Oscar. 15 wins & 60 nominations total","Poster":"https://m.media-
amazon.com/images/M/MV5BNWE5MGI3MDctMmU5Ni00YzI2LWEzMTQtZGIyZDA5MzQzNDBhXkEyXkFqcGc@._V1_SX300.jpg","Ratings":[{"Source":"Internet Movie Database","Value":"7.6/10"},{"Source":"Rotten Tomatoes","Value":"85%"},
{"Source":"Metacritic","Value":"67/100"}],"Metascore":"67","imdbRating":"7.6","imdbVotes":"797,474","imdbID":"tt3896198","Type":"movie","DVD":"N/A","BoxOffice":"$389,813,101","Production":"N/A","Website":"N/A","Res
ponse":"True"}
```

RESULT

Upon running the application, the initial interface displays a search box labeled "Enter movie name..." along with a Search button. When a user enters a movie name and clicks the button, the application sends a request to the index.js backend. The backend fetches movie data from the OMDb API and sends the results to the React frontend. The React frontend then dynamically displays a collection of movie cards in a responsive grid layout. Each card shows the movie's poster image, title, release year, and a link to watch the trailer. If poster data is missing, a default placeholder image is shown to maintain a consistent visual appearance




MovieFlick

Superman Search

Batman v Superman: Dawn of Justice
2016
Watch Trailer

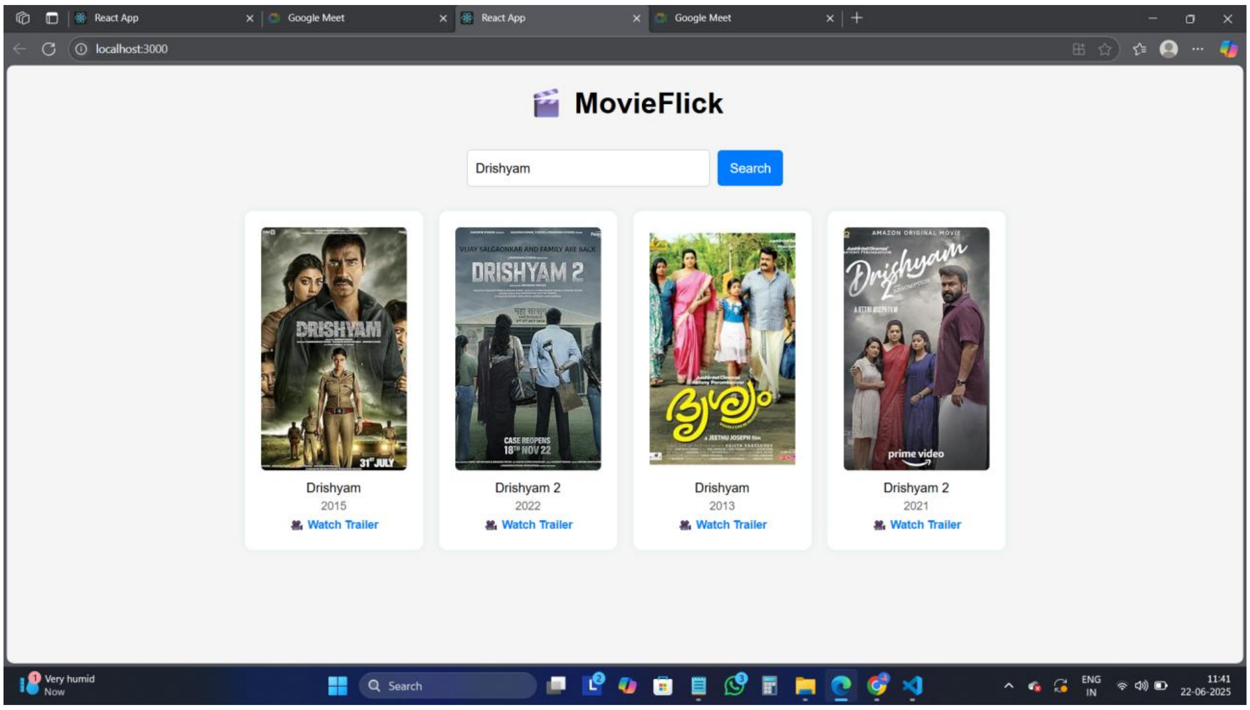
Superman Returns
2006
Watch Trailer

Superman
1978
Watch Trailer

Superman II
1980
Watch Trailer

Superman III
1983
Watch Trailer

Batman v Superman: Dawn of Justice (Ultimate Edition)
2016
Watch Trailer


MovieFlick

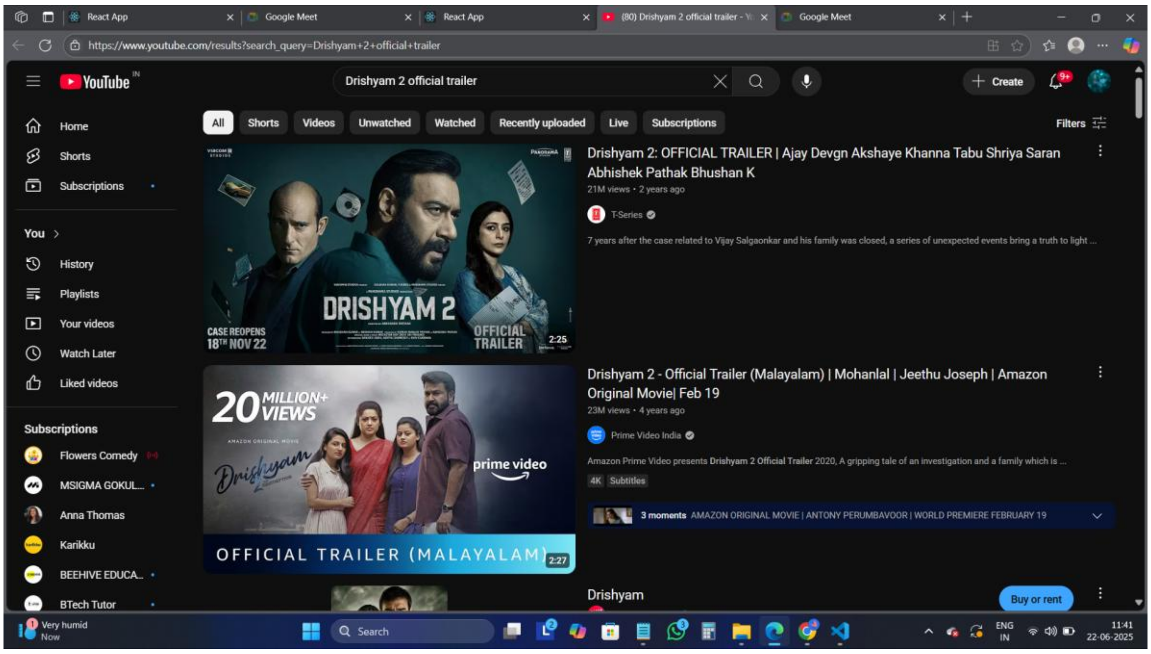
Drishyam Search

Drishyam
2015
Watch Trailer

Drishyam 2
2022
Watch Trailer

Drishyam
2013
Watch Trailer

Drishyam 2
2021
Watch Trailer



https://www.youtube.com/results?search_query=Drishyam+2+official+trailer

Drishyam 2 official trailer

Drishyam 2: OFFICIAL TRAILER | Ajay Devgn Akshaye Khanna Tabu Shriya Saran Abhishek Pathak Bhushan K
21M views • 2 years ago
T-Series
7 years after the case related to Vijay Salgaonkar and his family was closed, a series of unexpected events bring a truth to light ...

Drishyam 2 - Official Trailer (Malayalam) | Mohanlal | Jeethu Joseph | Amazon Original Movie| Feb 19
23M views • 4 years ago
Prime Video India
Amazon Prime Video presents **Drishyam 2 Official Trailer** 2020, A gripping tale of an investigation and a family which is ...
4K Subtitles
3 moments AMAZON ORIGINAL MOVIE | ANTONY PERUMBAVOOR | WORLD PREMIERE FEBRUARY 19

Drishyam

****