# MovieFlix Dashboard

MovieFlix is a movie information dashboard built with a React.js frontend and Node.js backend.

It integrates with an external movie API (e.g., OMDb API) to fetch comprehensive movie details.

It provides movie search functionality, detailed analytics, and a smooth user experience for both regular users and admins.

## Features

🎬 Core Features

Movie Search: Search movies by title using the OMDb API

\*Detailed Movie Information: View detailed movie information such as plot, cast, director, ratings, etc.

\*Advanced Filtering: Filter movies by genre, year, rating, and other criteria

\*Sorting Options: Sort movies by rating, year, runtime, or title

\*Responsive Design: Optimized for all devices including mobile, tablet, and desktop

\*Interactive Charts & Graphs: Visualize movie data through various charts like genre distribution, runtime trends, etc.

\*Error Handling: Graceful error management with user-friendly messages

##🔐 Authentication & Security

\*JWT-based authentication

\*Protected routes for user-specific content

\*Role-based access control (for Admin features)

##📊 Analytics Dashboard

\*Genre Distribution: Visualize movie genre breakdown in a pie chart

\*Average Ratings by Genre: Bar chart showing ratings across genres

\*Runtime Trends: Line chart showing the average runtime by release year

\*Movie Statistics: Aggregate data insights

## Tech Stack

@Frontend

\*React.js - UI framework

\*Axios - HTTP client for API requests

\*Chart.js - Data visualization

\*Tailwind CSS - Styling framework

\*React Router - Routing library

\*Context API - Global state management

@Backend

\*Node.js - Server-side runtime

\*Express.js - Web framework

\*MongoDB - Database (with MongoDB Compass)

\*JWT - JSON Web Token authentication

\*bcrypt - Password hashing

\*cors - CORS middleware

\*helmet - Security middleware

## External APIs & Services

\*OMDb API - Movie data provider

\*MongoDB Compass - Cloud database for backend data storage

##Project Structure

movieflix-dashboard/

├── README.md

├── package.json

├── .env.example

├── .gitignore

├── frontend/

│ ├── package.json

│ ├── public/

│ ├── src/

│ │ ├── components/

│ │ ├── pages/

│ │ ├── services/

│ │ ├── styles/

│ │

├── backend/

│ ├── package.json

│ ├── server.js

│ ├── config/

│ ├── models/

│ ├── routes/

│ └── controllers/

└── docs/

## Installation & Setup

@Prerequisites

\*Node.js (v16 or higher)

\*MongoDB (either local or MongoDB Compass)

\*OMDb API Key (obtainable from OMDb API)

1. Clone the Repository

git clone https://github.com/khaja1520/movieflix.git

cd movieflix

2. Environment Configuration

Create a .env file in the root directory based on the example below:

# Server Configuration

PORT=5000

NODE\_ENV=development

# Database Configuration

MONGODB\_URI=mongodb://localhost:27017/movieflix

# If using MongoDB Atlas, use the connection string from your Atlas cluster

# OMDb API Key

OMDB\_API\_KEY=your\_omdb\_api\_key\_here

# JWT Configuration

JWT\_SECRET=your\_jwt\_secret\_key\_here

JWT\_EXPIRE=7d

# Frontend URL (for CORS)

FRONTEND\_URL=http://localhost:3000

# Cache Configuration

CACHE\_EXPIRY\_HOURS=24

3. Backend Setup

Navigate to the backend folder and install dependencies:

cd backend

npm install

Start the backend server:

npm run dev

The backend will run at http://localhost:5000.

4. Frontend Setup

In a new terminal window, navigate to the frontend folder and install dependencies:

cd frontend

npm install

Start the frontend development server:

npm start

The frontend will run at http://localhost:3000.

5. Database Setup

Local MongoDB: Ensure MongoDB is installed and running on your local machine.

MongoDB Atlas: If you're using MongoDB Atlas, update the MONGODB\_URI in your .env file with your connection string.

## API Endpoints

@Authentication

\*POST /api/auth/register - Register a new user

\*POST /api/auth/login - Log in to the application

\*GET /api/auth/profile - Get the profile of the logged-in user (protected route)

Movies

GET /api/movies - Search movies with optional query parameters:

?search=movie\_title - Search by title

?sort=rating - Sort by rating, year, or title

?filter=genre:Action - Filter by genre

?page=1&limit=10 - Pagination for large datasets

GET /api/movies/:id - Get detailed movie information by id

POST /api/movies/refresh - Refresh the movie cache (admin-only)

Statistics

GET /api/stats/genres - Genre distribution data

GET /api/stats/ratings - Average ratings across genres

GET /api/stats/runtime - Runtime trends across years

GET /api/stats/overview - General statistics overview

## Usage Guide

1. Getting Started

Register for a new user account or log in using demo credentials.

Use the search bar to find movies.

Apply filters and sorting options to narrow your search.

Click on movie titles to view detailed information.

2. Admin Features

Admins can log in to access:

Movie cache management

Data refresh options

Advanced statistics and insights

3. Data Visualization

View interactive charts and graphs in the Stats page:

Genre distribution

Rating analysis by genre

Runtime trends over the years

Deployment

Vercel (Frontend)

To deploy the frontend on Vercel:

# Install Vercel CLI

npm i -g vercel

# Deploy frontend

cd frontend

vercel --prod

Heroku (Backend)

To deploy the backend on Heroku:

# Install Heroku CLI and log in

heroku login

# Create a new Heroku app

heroku create movieflix-api

# Set environment variables on Heroku

heroku config:set MONGODB\_URI=your\_mongodb\_atlas\_uri

heroku config:set OMDB\_API\_KEY=your\_omdb\_api\_key

heroku config:set JWT\_SECRET=your\_jwt\_secret\_key

# Deploy to Heroku

git subtree push --prefix backend heroku main

Railway (Full Stack Deployment)

Railway can deploy the full-stack app automatically when linked with your GitHub repository. Set up the environment variables and deploy the project using Railway's build system.

Environment Variables

Required

OMDB\_API\_KEY: Your OMDb API key

MONGODB\_URI: MongoDB connection string

JWT\_SECRET: JWT secret key for authentication

Optional

PORT: Server port (default: 5000)

CACHE\_EXPIRY\_HOURS: Cache expiry time in hours (default: 24)

FRONTEND\_URL: Frontend URL for CORS

Scripts

Backend

npm start # Start production server

npm run dev # Start development server with nodemon

npm run test # Run backend tests

npm run lint # Run linting

Frontend

npm start # Start frontend server

npm run build # Build for production

npm run test # Run frontend tests

Contributing

Fork the repository

Create a new branch (git checkout -b feature/my-feature)

Commit your changes (git commit -m 'Add feature')

Push to the branch (git push origin feature/my-feature)

Create a Pull Request

License

This project is licensed by Khaja Moinudeen A K

Support

For issues, questions, or support:

Create an issue on GitHub

Email us at [khajamoinudeen1574@gmail.com]

Acknowledgments

OMDb API for movie data

React.js for building the frontend

Express.js for the backend

MongoDB Atlas for cloud database hosting

## Screenshots

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A close-up of a white background

AI-generated content may be incorrect.

A close-up of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A white screen with a blue border

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.