C.V. RAMAN GLOBAL UNIVERSITY

BHUBANESWAR , ODISHA

A CASE STUDY FOR :

WEB TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DOMinators :

1) ARPEET BARIK - 2301020228
2)CH SAIPRIYA PATRA - 2301020205
3)DEEPSIKHA PATRA - 2301020242
4)LIKUNA SWAIN - 2301020252
5)SATYA PRAKASH ROUT - 2301020206

TOPIC :-

CREATE A WEBSITE LIKE ALL TIME BOX OFFICE MOJO. THAT WILL CONTAINS A DATA FOR ABOUT 1000 HIGHEST GROSSING MOVIES LIST WITH DIRECTORS, PRODUCERS,ACTORS AND ALL STAR CAST CREW RELATED TO IT.AND ADD AN ADMIN DATA BASE INTO IT.
LOGIN PAGE AND SIGN UP PAGE MANDATORY.

INTRODUCTION

PROJECT OVERVIEW

This project is a full-stack movie database website inspired by Box Office Mojo.
It lists the top 1000 highest-grossing movies with detailed cast and crew information.
The system includes user authentication, admin control, and real-time data from the TMDB API.

A web portal showcasing **Top 1000 Highest-Grossing Movies**

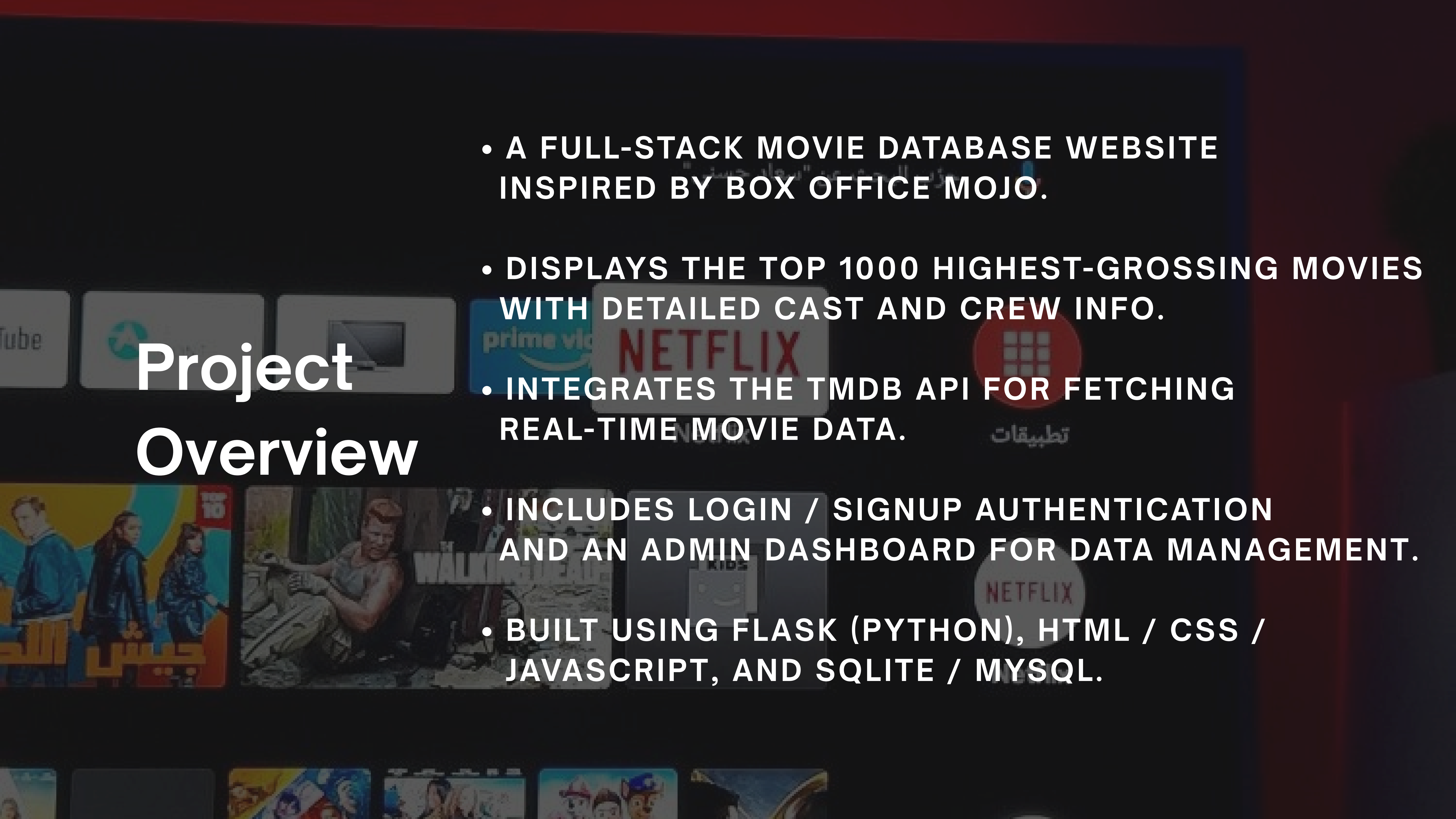
Displays **directors, producers, actors, and crew details**

Includes **Admin Dashboard** with **Login and Signup** pages

Built using **Flask (Python)**, **HTML / CSS / JavaScript**, **SQLite / MySQL**

Uses **TMDB API** to fetch and update real-time movie data

Provides a **modern, responsive UI** for an engaging user experience



Project Overview

- A FULL-STACK MOVIE DATABASE WEBSITE INSPIRED BY BOX OFFICE MOJO.
- DISPLAYS THE TOP 1000 HIGHEST-GROSSING MOVIES WITH DETAILED CAST AND CREW INFO.
- INTEGRATES THE TMDB API FOR FETCHING REAL-TIME MOVIE DATA.
- INCLUDES LOGIN / SIGNUP AUTHENTICATION AND AN ADMIN DASHBOARD FOR DATA MANAGEMENT.
- BUILT USING FLASK (PYTHON), HTML / CSS / JAVASCRIPT, AND SQLITE / MYSQL.

System Overview

FRONTEND

Handles user interaction and movie display

BACKEND

Flask framework managing logic and API integration

DATABSE

Stores movies and users securely

TMD API

Fetches updated movie data automatically

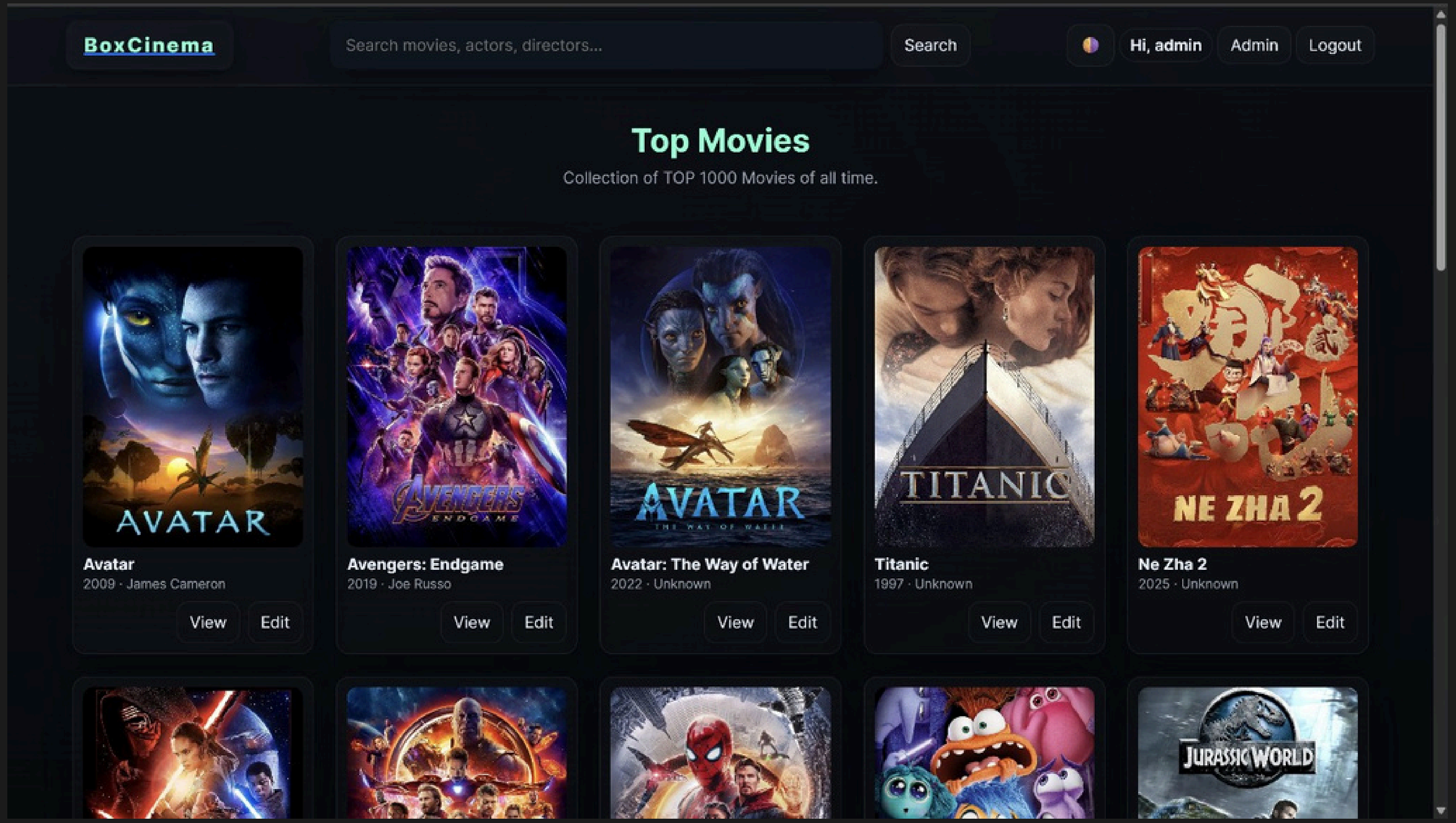
ADMIN

Manages movie list and user records

Frontend Overview

- Elegant Glass-morphism UI inspired by Netflix for a cinematic user experience
- Dynamic movie cards rendered in real-time using data fetched from the Flask API
- Instant search bar and popup windows displaying detailed movie information
- Fully responsive layout optimized for mobile, tablet, and desktop devices
- Smooth Dark / Light theme toggle with modern transition and hover animations

HOMEPAGE SHOWING ALL TOP MOVIES



Top Movies

Collection of TOP 1000 Movies of all time.

Avatar
2009 · James Cameron

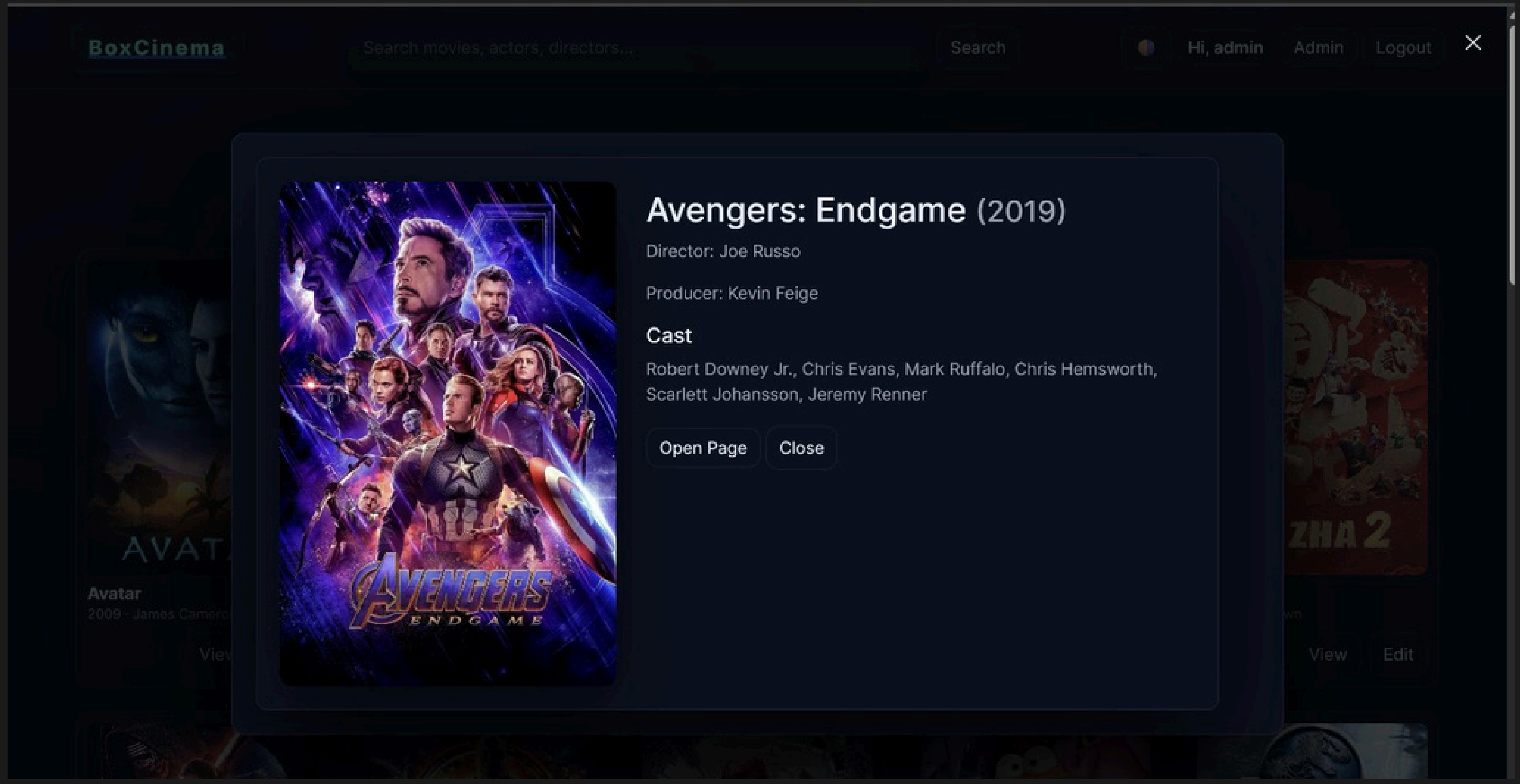
Avengers: Endgame
2019 · Joe Russo

Avatar: The Way of Water
2022 · Unknown

Titanic
1997 · Unknown

Ne Zha 2
2025 · Unknown

Single Movie Details popup



Avengers: Endgame (2019)

Director: Joe Russo

Producer: Kevin Feige

Cast

Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth, Scarlett Johansson, Jeremy Renner

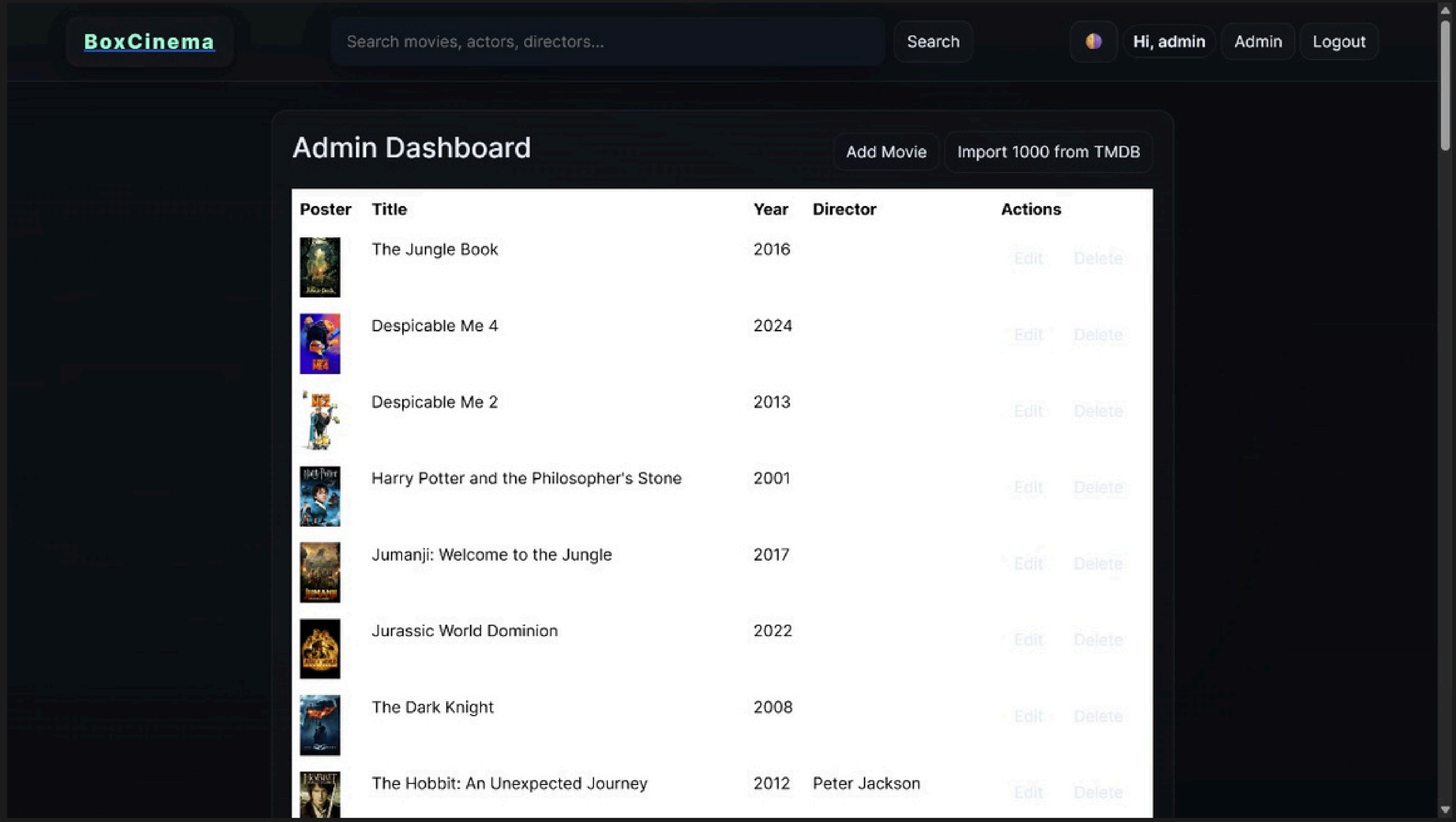
Frontend Tools & Technologies

Purpose	Technology
Structure	HTML5
Styling	Tailwind CSS + Custom CSS
Script	JavaScript (Vanilla)
Animation	Framer Motion / GSAP
Icons	Font Awesome / Lucide Icons
Template	Jinja2
Layout	Grid + Flexbox

Backend Overview

- Developed using Flask (Python) as the core backend framework for logic handling
- Connects the Frontend UI ↔ TMDB API ↔ Database to manage data flow efficiently
- Handles user authentication, movie data import, and admin-level access controls
- Provides RESTful JSON endpoints that serve movie data dynamically to the frontend
- Ensures secure and optimized data processing for reliable performance

Admin Dashboard Layout



Admin Dashboard

Poster	Title	Year	Director	Actions
	The Jungle Book	2016		Edit Delete
	Despicable Me 4	2024		Edit Delete
	Despicable Me 2	2013		Edit Delete
	Harry Potter and the Philosopher's Stone	2001		Edit Delete
	Jumanji: Welcome to the Jungle	2017		Edit Delete
	Jurassic World Dominion	2022		Edit Delete
	The Dark Knight	2008		Edit Delete
	The Hobbit: An Unexpected Journey	2012	Peter Jackson	Edit Delete

Backend Tools & Technologies

Purpose	Technology
Framework	Flask
Database	SQLite / MySQL
ORM	SQLAlchemy
Auth	Flask-Login / Sessions
API	TMDB REST API
Config	python-dotenv
Template Engine	Jinja2

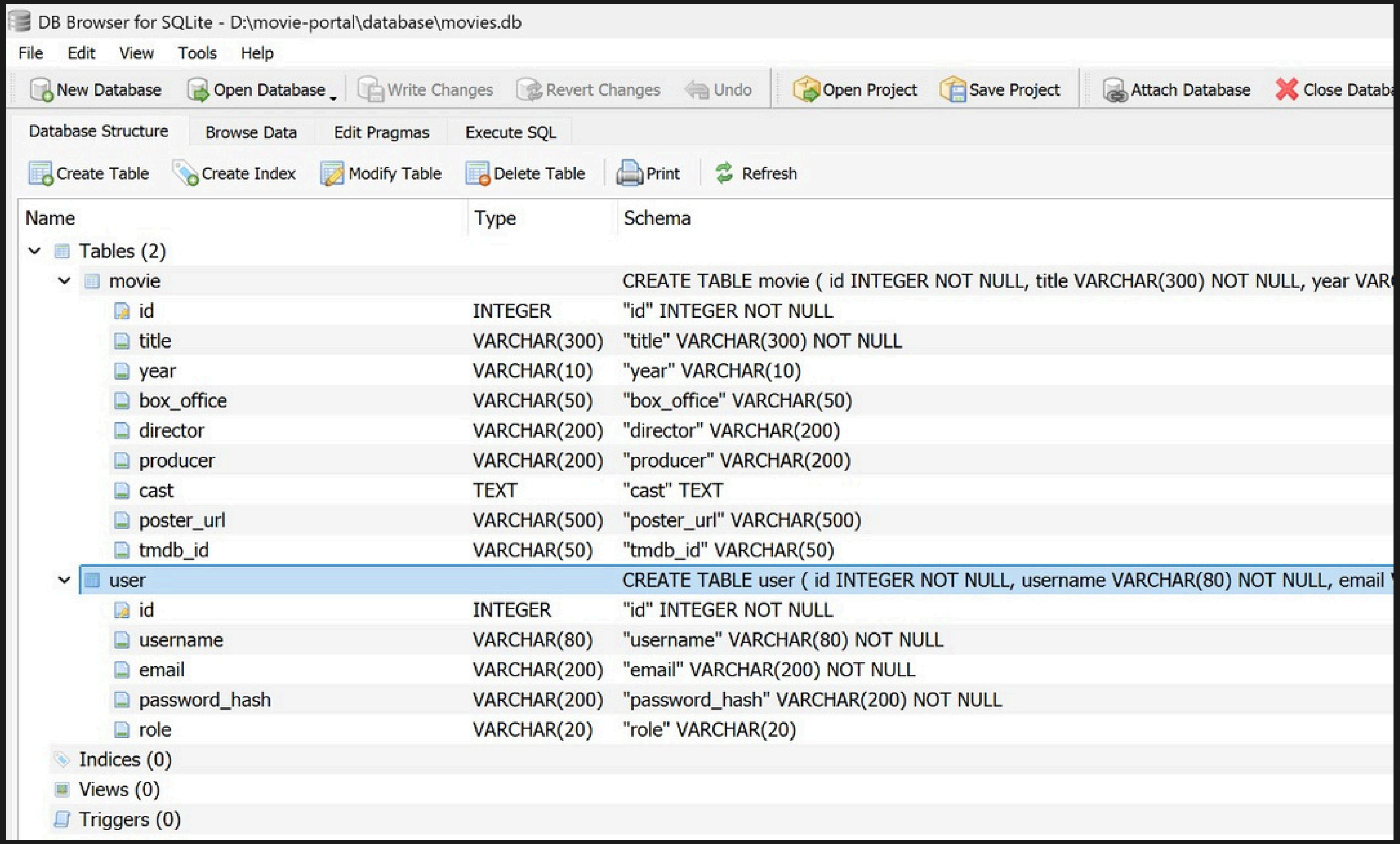
Database & Admin Panel

- Designed a structured database to store both user and movie information efficiently
- User Table: id, username, password (hashed), is_admin – for authentication & access control
- Movie Table: id, title, year, director, description, image_url – for complete movie details

Admin Panel Features:

- Import and update movies directly via the TMDB API
- View, edit, or delete movie entries stored in the database
- Manage user privileges and monitor overall database activity
- Ensures secure data storage and easy scalability for future enhancements

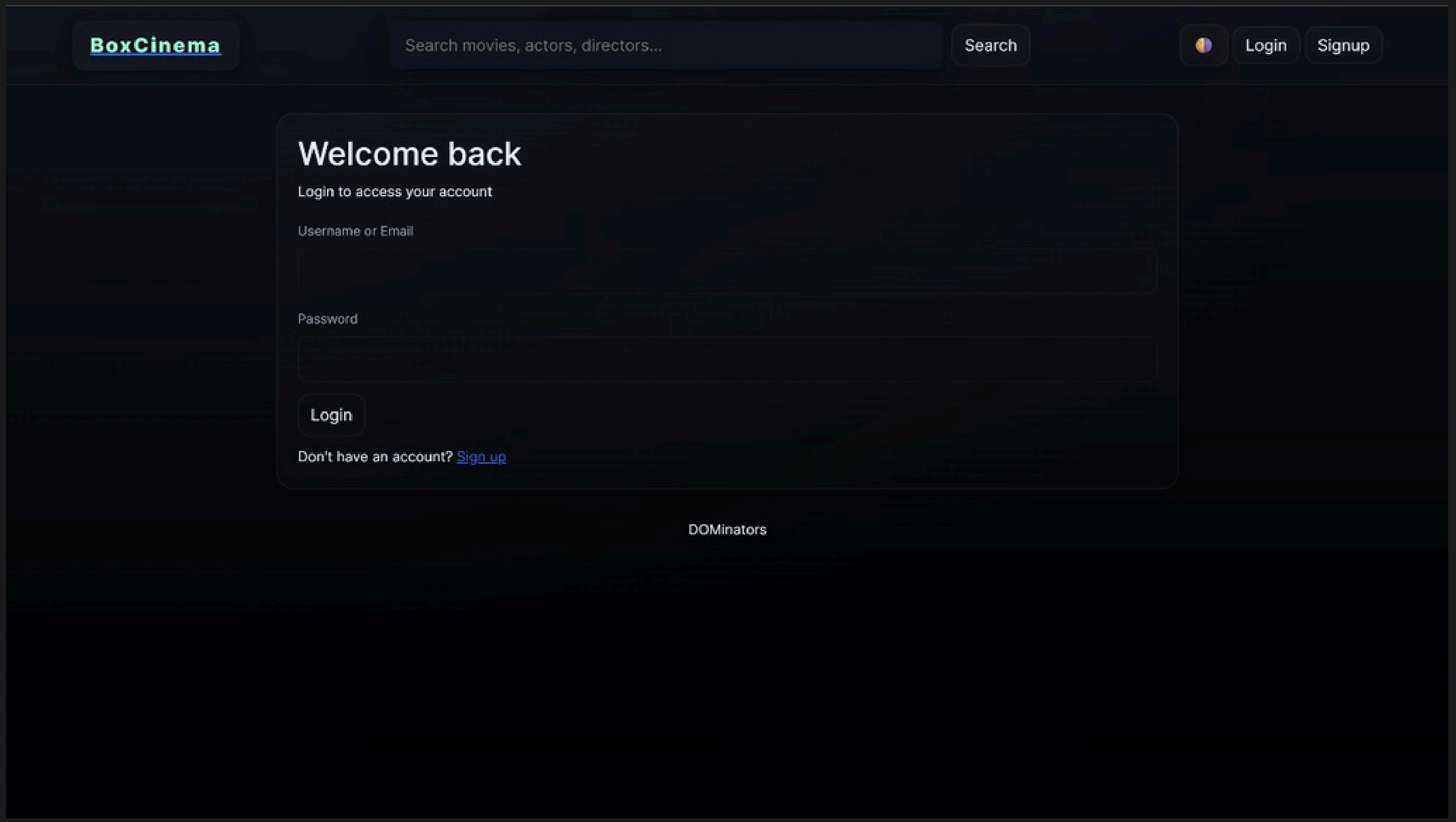
Database Query Structure



Name	Type	Schema
Tables (2)		
movie		CREATE TABLE movie (id INTEGER NOT NULL, title VARCHAR(300) NOT NULL, year VAR
id	INTEGER	"id" INTEGER NOT NULL
title	VARCHAR(300)	"title" VARCHAR(300) NOT NULL
year	VARCHAR(10)	"year" VARCHAR(10)
box_office	VARCHAR(50)	"box_office" VARCHAR(50)
director	VARCHAR(200)	"director" VARCHAR(200)
producer	VARCHAR(200)	"producer" VARCHAR(200)
cast	TEXT	"cast" TEXT
poster_url	VARCHAR(500)	"poster_url" VARCHAR(500)
tmdb_id	VARCHAR(50)	"tmdb_id" VARCHAR(50)
user		CREATE TABLE user (id INTEGER NOT NULL, username VARCHAR(80) NOT NULL, email
id	INTEGER	"id" INTEGER NOT NULL
username	VARCHAR(80)	"username" VARCHAR(80) NOT NULL
email	VARCHAR(200)	"email" VARCHAR(200) NOT NULL
password_hash	VARCHAR(200)	"password_hash" VARCHAR(200) NOT NULL
role	VARCHAR(20)	"role" VARCHAR(20)
Indices (0)		
Views (0)		
Triggers (0)		

Authentication Pages

LOGIN PAGE



BoxCinema

Search movies, actors, directors...

Search

Login

Signup

Welcome back

Login to access your account

Username or Email

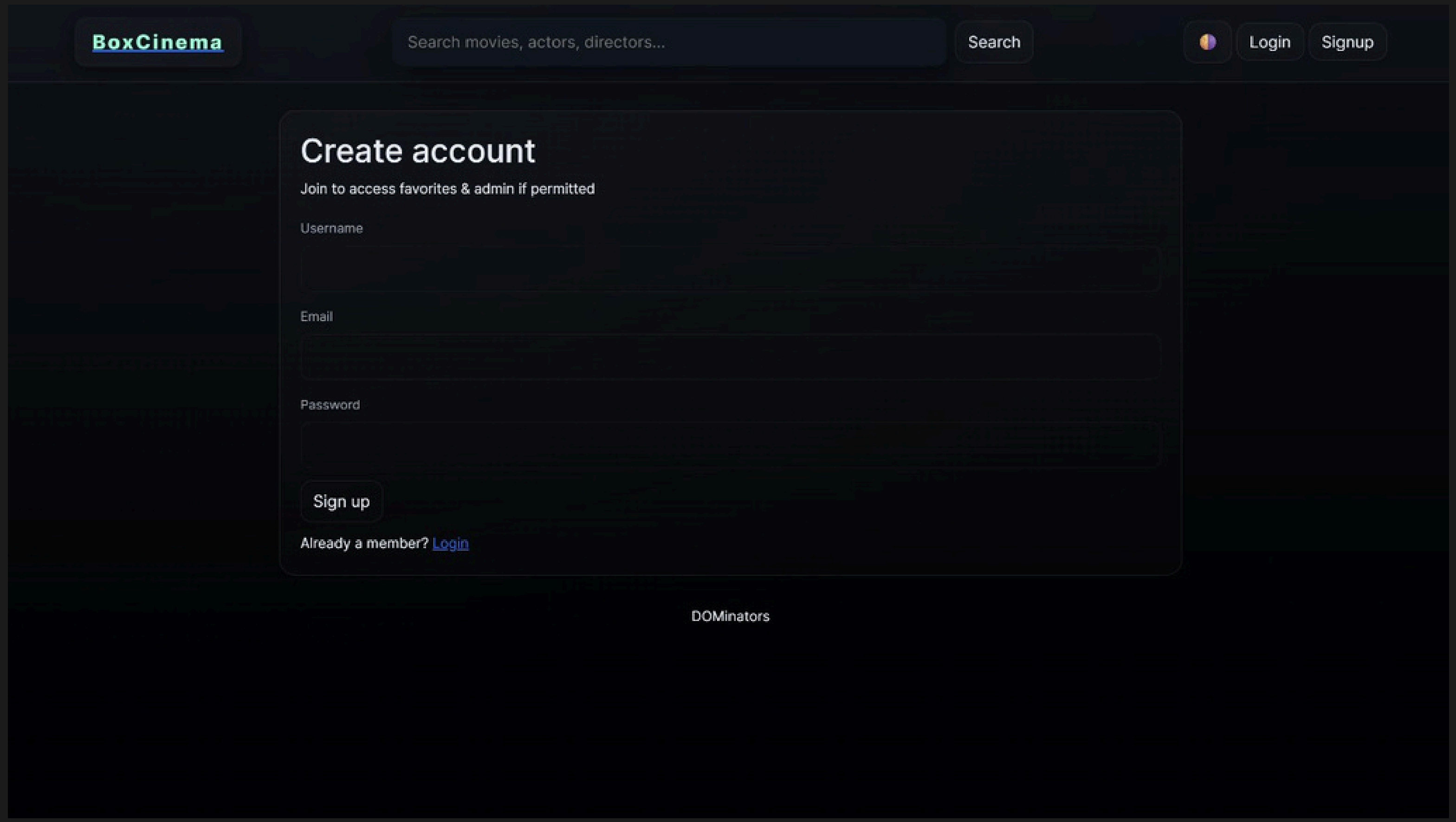
Password

Login

Don't have an account? Sign up

DOMinators

SIGNUP PAGE



BoxCinema

Search movies, actors, directors...

Search

Login

Signup

Create account

Join to access favorites & admin if permitted

Username

Email

Password

Sign up

Already a member? Login

DOMinators

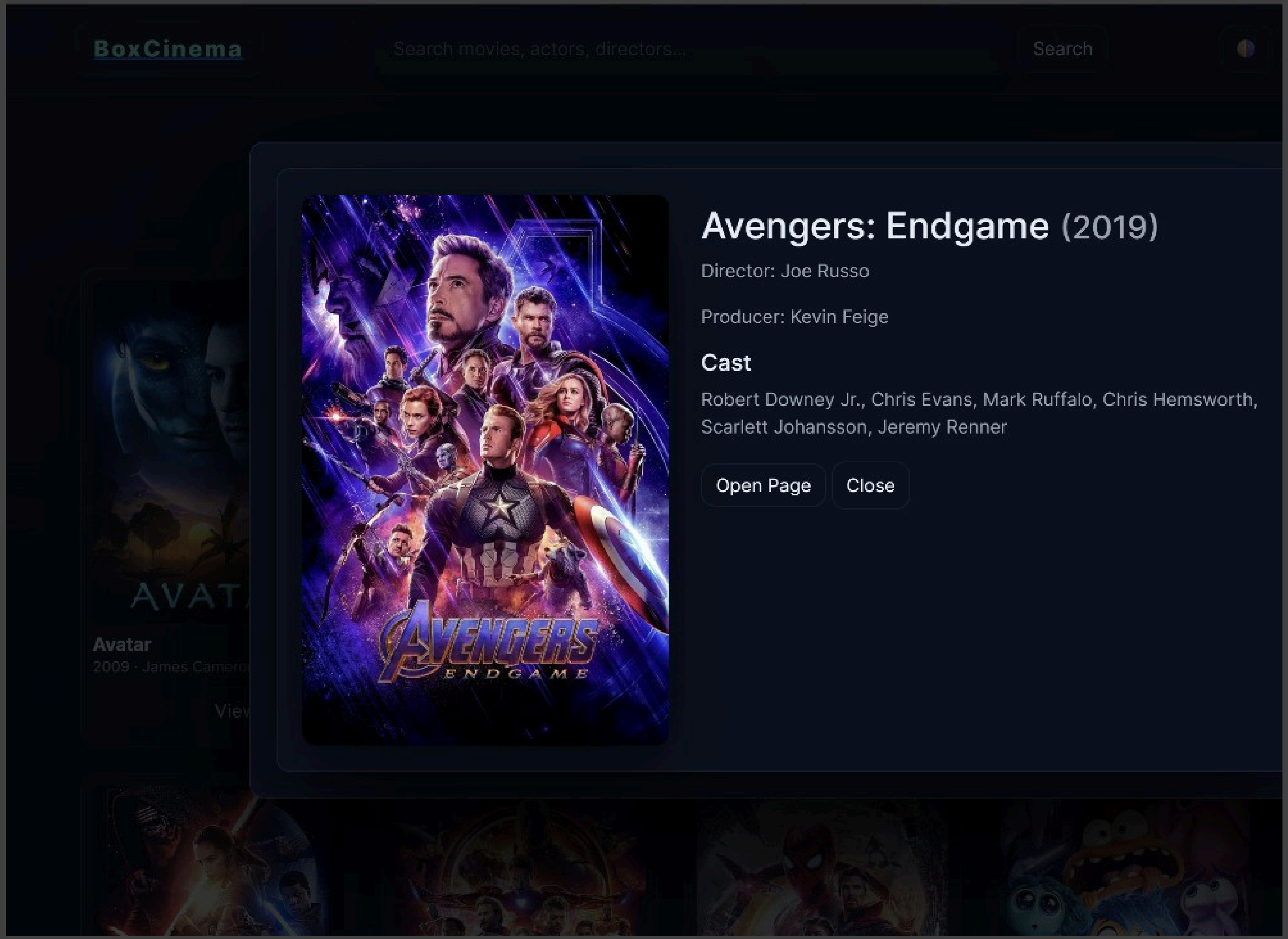
- **Secure Login:** Uses hashing to validate passwords securely.
- **Access Control:** Redirects users instantly based on their role (Admin/User).

- **Security Focus:** Stores new passwords securely using hashing protocols.
- **User Experience:** Features smooth UI transitions for a better sign-up flow.

Challenges & Solutions

Technical challenges—including API throttling, large dataset management, and UI performance degradation—were resolved through pagination, retry mechanisms, and rendering optimization. System stability and security were ensured via password hashing and session management.

Challenge: TMDB API request limits	**Solution:** Implemented per-page fetching and added delay handling
Challenge: Managing data for 1000+ movies	**Solution:** Used lazy loading and pagination to improve speed and performance
Challenge: Ensuring secure authentication	**Solution:** Applied password hashing and session-based login management
Challenge: API connection failures or timeouts	**Solution:** Added a retry mechanism and error-handling logic for stability
Challenge: UI performance and responsiveness	**Solution:** Optimized **CSS blur effects** and **JavaScript animations** for smoother transitions



Avengers: Endgame (2019)

Director: Joe Russo

Producer: Kevin Feige

Cast

Robert Downey Jr., Chris Evans, Mark Ruffalo, Chris Hemsworth, Scarlett Johansson, Jeremy Renner

Results & Conclusion

- Successfully developed a full-stack movie portal
- Displays 1000+ real-time movies with complete data
- Secure Login/Signup and Admin Panel integration
- Responsive, modern, and data-driven design
- Demonstrates complete frontend + backend integration

Thank You – Team DOMinators