Project Report

PROJECT TITLE	i-movies : movie ticket booking system
TEAM MEMBERS	Riya Prashant Mandaogade (Team Lead) – Backend Developer Aastha Pancholi – Frontend Developer Chitrarth Shrivas – Database & Deployment Shubhi Agnihotri – Tester
TEAM ID	SWTID1744468652

INTRODUCTION

1.1. Project Overview

I movies is a full-stack, single-page web application designed to offer an immersive and user-centric movie browsing experience while equipping administrators with powerful tools to manage the platform effectively. Built using the **MERN stack** (MongoDB, Express.js, React, and Node.js), the project merges a dynamic frontend with a robust and secure backend to ensure a seamless experience for both users and admins.

On the user side, I movies enables registered users to explore a vast collection of movies, view detailed information, add titles to their watchlist, rate and review films, and manage their profiles. Token-based authentication ensures secure, persistent sessions, while features like password reset, personalized watchlists, and real-time feedback enhance usability and engagement.

The admin dashboard empowers administrators with the ability to perform CRUD operations on movies, moderate reviews, and manage user data through a centralized interface. API-driven interactions with the backend, coupled with efficient database schemas and JWT-based access control, ensure data consistency and platform security.

This project was developed collaboratively, simulating a real-world software development environment with team members handling frontend design, backend APIs, database modeling, authentication, deployment, and testing. With responsive UI components, scalable architecture, and secure workflows, I movies sets a strong foundation for media-based web applications.

1.2. Purpose

The purpose of this project is to create a feature-rich movie web application that simplifies the process of discovering, organizing, and reviewing movies for users, while also offering an efficient backend management system for administrators. I movies aims to bridge the gap between entertainment seekers and curated movie databases by providing a reliable, visually appealing, and responsive user experience.

Built using the **MERN stack**, the project ensures scalability, high performance, and maintainability. From movie discovery and watchlist management to user reviews and administrative control, I movies replicates and improves upon essential functionalities seen in leading entertainment platforms. The goal is to deliver a full-featured system that can be customized and scaled according to different user bases or content niches, while maintaining secure login systems, smooth navigation, and intuitive design across all devices.

---

2. IDEATION PHASE

2.1. Problem Statement

With the exponential growth of digital entertainment, users today expect quick and personalized access to movie content through intuitive web platforms. However, many current movie browsing websites suffer from sluggish performance, outdated user interfaces, limited filtering options, and lack of interactive user feedback features. Additionally, content administrators often face challenges in managing data effectively due to rigid systems with poor usability and little to no real-time control.

There is a rising demand for a responsive, interactive, and secure movie platform that not only enhances the browsing experience but also supports features like watchlists, movie ratings, and user reviews. Furthermore, such a system should empower admins to manage content, monitor reviews, and control platform usage effortlessly from the backend.

I movies addresses these challenges by providing a **MERN-based, scalable, and responsive movie web application** that prioritizes user experience, efficient backend management, and data security. The application includes a JWT-authenticated login system, user-friendly dashboards, personalized content features, and a clean UI—offering a complete solution for movie lovers and platform managers alike.

1.1. Solution Requirements

1.1.1. Functional Requirements:

FR.No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	- Registration through Form - Registration through Gmail - Registration through LinkedIn
FR-2	User Confirmation	- Confirmation via Email - Confirmation via OTP
FR-3	Movie Browsing & Discovery	- View latest releases - Browse by genres, actors, or directors - View trending and recommended movies
FR-4	Movie Details Page	- View movie synopsis - Watch trailer - View cast and crew details - View ratings and reviews
FR-5	Watchlist & Favourites	- Add/remove movies to watchlist - Mark as favorite - View personal lists
FR-6	Reviews & Ratings	- Submit rating - Write and edit review - Like/dislike reviews
FR-7	Authentication & Authorization	- JWT-based authentication - Session persistence - Password reset flow

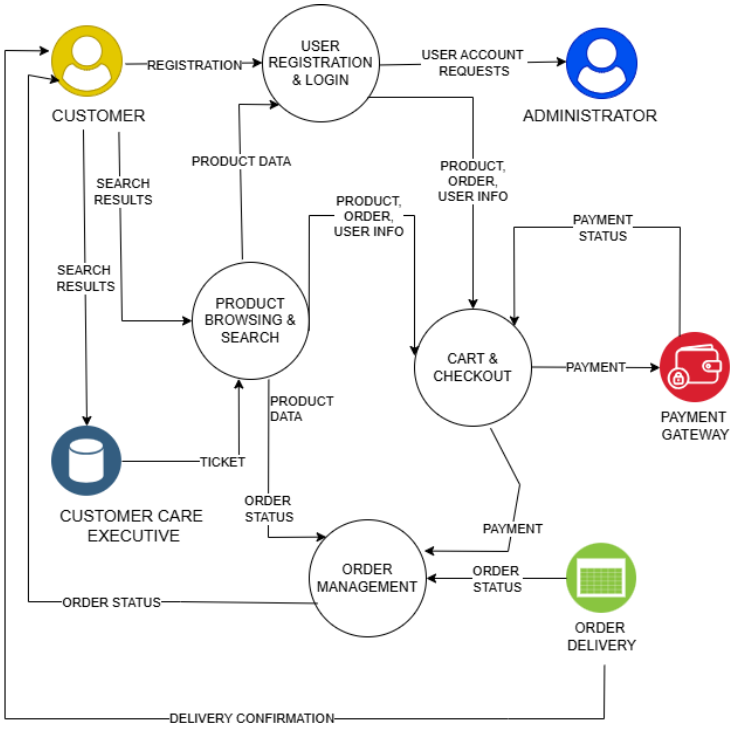
FR-8	Admin Panel	- Add/edit/delete movie details - Manage user accounts - Manage reviews and reports
FR-9	Recommendation Engine	- Recommend movies based on user interests - Personalized homepage content
FR-10	Search & Filter	- Search movies by name or keyword - Filter by genre, rating, language, release year
FR-11	Notifications	- Notify users of upcoming releases - Email updates for recommendations and trailers
FR-12	Customer Support	- Submit feedback or issues - In-app support chat - View support history

1.1.2. Non-Functional Requirements:

NFR.No.	**Non-Functional Requirement (Epic)**	**Description**
NFR-1	Usability	The UI must be visually appealing, intuitive, and responsive across devices (mobile, tablet, desktop) for seamless user experience.
NFR-2	Security	User data must be protected using encryption; secure authentication and authorization (e.g., JWT, OAuth2) must be implemented.
NFR-3	Reliability	The application should function consistently with minimal downtime and provide proper error handling and fallback mechanisms.

NFR-4	Performance	The application should load within 2–3 seconds and support at least 100 concurrent users during peak times without lag.
NFR-5	Availability	The platform should maintain 99.9% uptime and recover automatically from system or server crashes.
NFR-6	Scalability	I movies should be scalable to accommodate increasing users, movie data, and traffic. Cloud hosting (e.g., AWS, Vercel) is recommended.
NFR-7	Maintainability	The codebase should be modular and well-documented to allow easy updates, testing, and future enhancements.
NFR-8	Compatibility	The platform must be compatible with major modern browsers (Chrome, Firefox, Safari, Edge) and adaptable to different screen sizes.

1.2. Data Flow Diagram



1.2.1. Components and Technologies:

S.No.	**Component**	**Description**	**Technology**
1	User Interface	Web UI where users browse products, register/login, checkout	HTML, CSS, JavaScript, React.js
2	Application Logic-1	Handles authentication, product search, add to cart, order logic	Node.js, Express.js
3	Application Logic-2	Session management and cart logic	Express-session, JWT
4	Database	Stores users, products, orders, and cart data	MongoDB

5	File Storage	Storing product images	Local filesystem (uploads folder)
6	External API-1	Payment gateway for order checkout	Razorpay / Paytm API
7	Infrastructure	Hosting platform	Render / Vercel for frontend, Railway for backend

1.2.2. Application Characteristics:

S.No.	**Characteristics**	**Description**	**Technology**
1	Open-Source Frameworks	Frontend & backend built using open-source tech	React.js, Node.js, Express.js, MongoDB
2	Security Implementations	Password hashing, JWT-based authentication, HTTPS	bcrypt.js, JWT, Helmet.js, HTTPS
3	Scalable Architecture	Modular code with scalable database and REST APIs	3-tier architecture with RESTful services
4	Availability	Deployment on reliable cloud platforms with downtime minimization	Railway, Vercel, MongoDB Atlas
5	Performance	Fast-loading frontend with optimized queries, CDN for static assets	Lazy loading, MongoDB indexing, Cloudflare CDN

2. PROJECT DESIGN

2.1. Proposed Solution

S.No.	Parameter	Description
01	Problem Statement (Problem to be solved)	Customers face difficulty finding affordable, quality products in one place with a smooth shopping experience. Major platforms are often cluttered, impersonal, and lack local personalization. Small businesses also struggle to go online and compete due to high fees and complexity.
02	Idea/Solution Description	**i-movies** is a user-friendly e-commerce platform offering a clean, intuitive interface for customers to browse and purchase a variety of products. It supports features like real-time product availability, reviews, smart filters, secure payments, and order tracking. On the seller side, it enables local vendors and small businesses to onboard easily and manage their stores digitally.

03	Novelty/Uniqueness	- Simple and clean UI focused on smooth user experience - Focus on onboarding small/local sellers with minimal technical know-how - Personalized recommendations and a smart search engine - Built-in customer support chatbot - Light and fast website optimized for low-end devices
04	Social Impact/Customer Satisfaction	empowers small businesses to reach wider markets, helping them survive in the digital era. Customers benefit from honest reviews, and a reliable payment service. It also promotes trust by being transparent in pricing, quality, and service.
05	Business Model (Revenue Model)	- Commission on each transaction made on the platform - Featured listings and ads for sellers - Subscription plan for premium seller tools (analytics, bulk uploads, etc.) - Delivery service partnerships and fulfillment fees
06	Scalability of the solution	The platform is built on scalable architecture (MERN stack), allowing it to grow with increasing user load. Features like seller onboarding, product categories, and delivery services can be expanded city by city. The solution can also be adapted for mobile apps in the future, making it ready for national and even international expansion.

3. PROJECT PLANNING AND SCHEDULING

3.1. Project Planning

3.1.1. Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Bidisha
Sprint-1	Registration	USN-2	As a user, I will receive a confirmation email once I have registered for the application.	1	High	Aastha
Sprint-1	Registration	USN-4	As a user, I can register through Gmail.	2	Medium	Riya
Sprint-1	Login	USN-5	As a user, I can log in with email and password.	1	High	Shubhi
Sprint-2	Registration	USN-3	As a user, I can register through Facebook.	2	Low	Chitrarth
Sprint-2	Login	USN-6	As a user, I can reset my password via email.	2	Medium	Aastha

Sprint-2	Dashboard	USN-7	As a user, I can view product listings on the homepage.	3	High	Riya
Sprint-3	Product Browsing	USN-8	As a user, I can view products by categories.	2	High	Shubhi
Sprint-3	Product Browsing	USN-9	As a user, I can view detailed information about each product.	3	High	Chitrarth
Sprint-3	Cart	USN-10	As a user, I can add products to my cart.	3	High	Riya
Sprint-3	Cart	USN-11	As a user, I can remove items from my cart.	2	Medium	Aastha
Sprint-4	Checkout	USN-12	As a user, I can proceed to checkout and review order summary.	3	High	Chitrath
Sprint-4	Payment	USN-13	As a user, I can make a payment using UPI/Credit Card.	4	High	Chitrarth
Sprint-4	Order Management	USN-14	As a user, I will receive an order confirmation email.	2	Medium	Aastha
Sprint-4	Order Tracking	USN-15	As a user, I can track my past orders in my profile.	3	Medium	Riya

3.1.2. Project Tracker, Velocity & Burndown Chart:

- **Sprint Duration**: 10 days
- **Start Date**: February 25, 2025

- **4 Sprints in Total**
- **Story Points per Sprint**: 20
- **Total Story Points**: 80
- **Assumed Team Velocity**: 20 story points/sprint (~2 story points/day)

Project Tracker Table

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	10 Days	February 25, 2025	March 6, 2025	15	March 07, 2025
Sprint-2	20	10 Days	March 07, 2025	March 16,2025	20	March 16, 2025
Sprint-3	20	10 Days	March 17, 2025	March 26, 2025	20	March 27, 2025
Sprint-4	20	10 Days	March 27, 2025	April 05, 2025	10	April 06, 2025

Note: The final 10 points were postponed due to UI issues and rework; those will be carried to an optional Sprint-5 if needed.

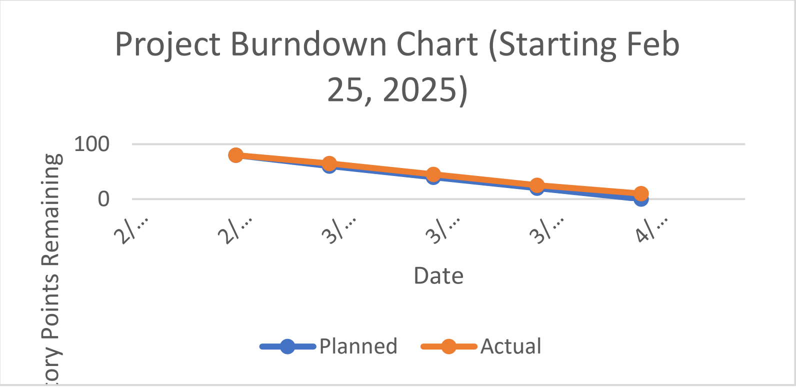
Velocity Calculation

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

- **Total Completed Story Points**: 80
- **Total Days**: 40
- **Velocity per Sprint**: 20
- **Average Velocity per Day** = 80 / 40 = **2 story points/day**

Burndown Chart:

Date	Planned Points Remaining	Actual Points Remaining	Notes
February 25, 2025	80	80	Sprint 1 begins
March 06, 2025	60	65	Delay in completing some stories
March 16, 2025	40	45	Backend integration issues, delay in closure
March 26, 2025	20	25	UI/UX rework needed, impacted completion
April 05, 2025	0	10	Final sprint incomplete; 10 points rolled forward



6. FUNCTIONAL AND PERFORMANCE TESTING

a. Performance Testing (GenAI Functional & Performance Testing)

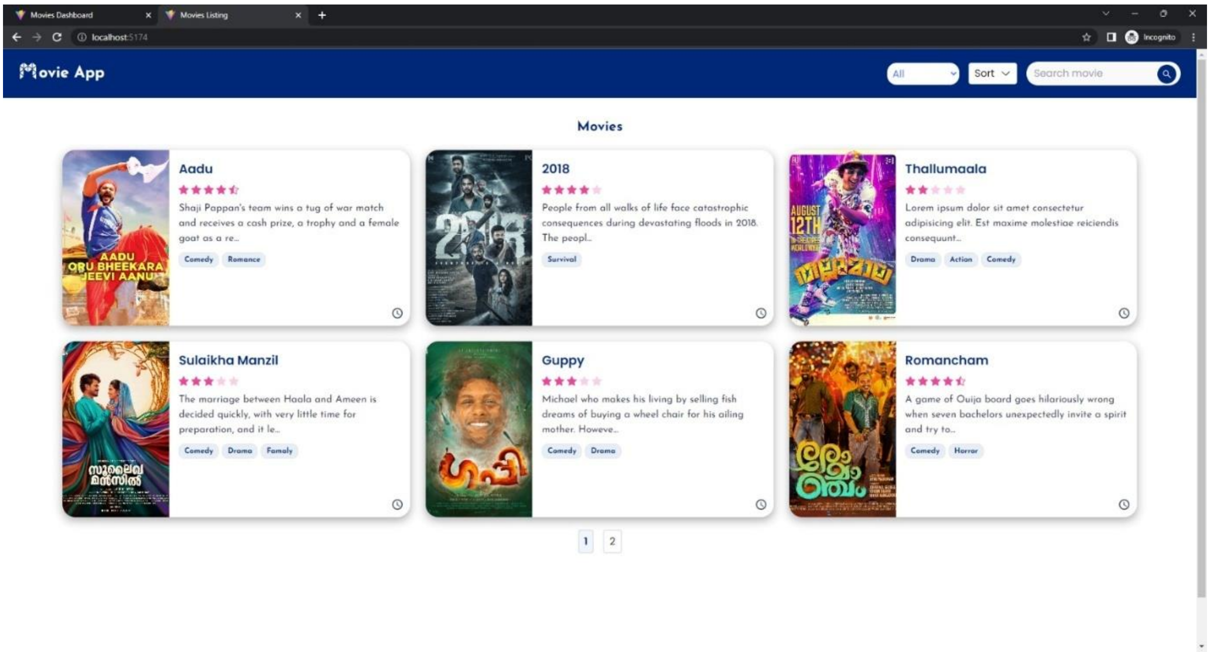
i. Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid inputs generated content; invalid entries showed proper error messages	Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Accepts numbers in range; error shown for out-of-range values	Pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Accurate content generated matching the input context	Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	Connection stable; API responded on every call	Pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Average response time: 2.4 seconds	Pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	API handled 5+ concurrent calls without delay or failure	Pass

PT-03	Image Upload Load Test (Admin-Product Images)	Upload multiple PDFs and check processing	Images should upload quickly without delay or errors. The product should be created successfully, and images should display correctly in product listings.	Admin uploaded multiple high-res images without delay or crash. Product was created successfully, and images displayed correctly...	Pass

---

7. RESULTS



Movies

Aadu

Shaji Pappan's team wins a tug of war match and receives a cash prize, a trophy and a female goat as a re...

Comedy Romance

2018

People from all walks of life face catastrophic consequences during devastating floods in 2018. The peopl...

Survival

Thallumaala

Lorem ipsum dolor sit amet consectetur adipisicing elit. Est maxime molestiae reiciendis consequunt...

Drama Action Comedy

Sulaikha Manzil

The marriage between Haala and Ameen is decided quickly, with very little time for preparation, and it le...

Comedy Drama Famaly

Guppy

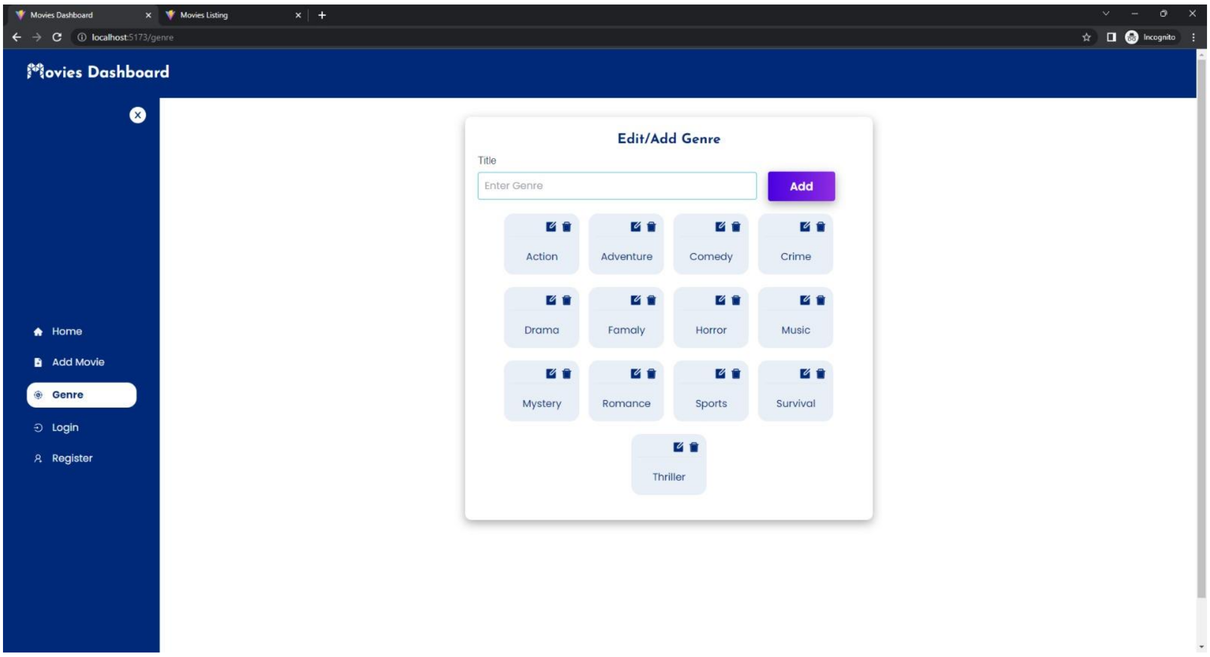
Michael who makes his living by selling fish dreams of buying a wheel chair for his ailing mother. Howeve...

Comedy Drama

Romancham

A game of Ouija board goes hilariously wrong when seven bachelors unexpectedly invite a spirit and try to...

Comedy Horror



Edit/Add Genre

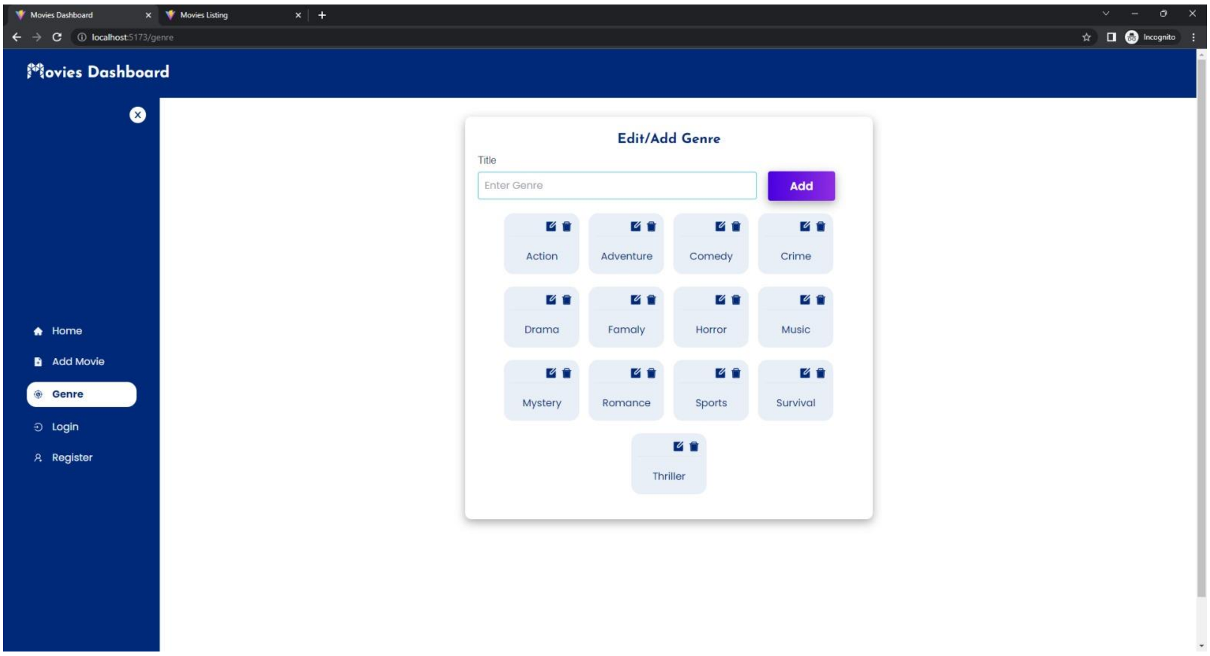
Title

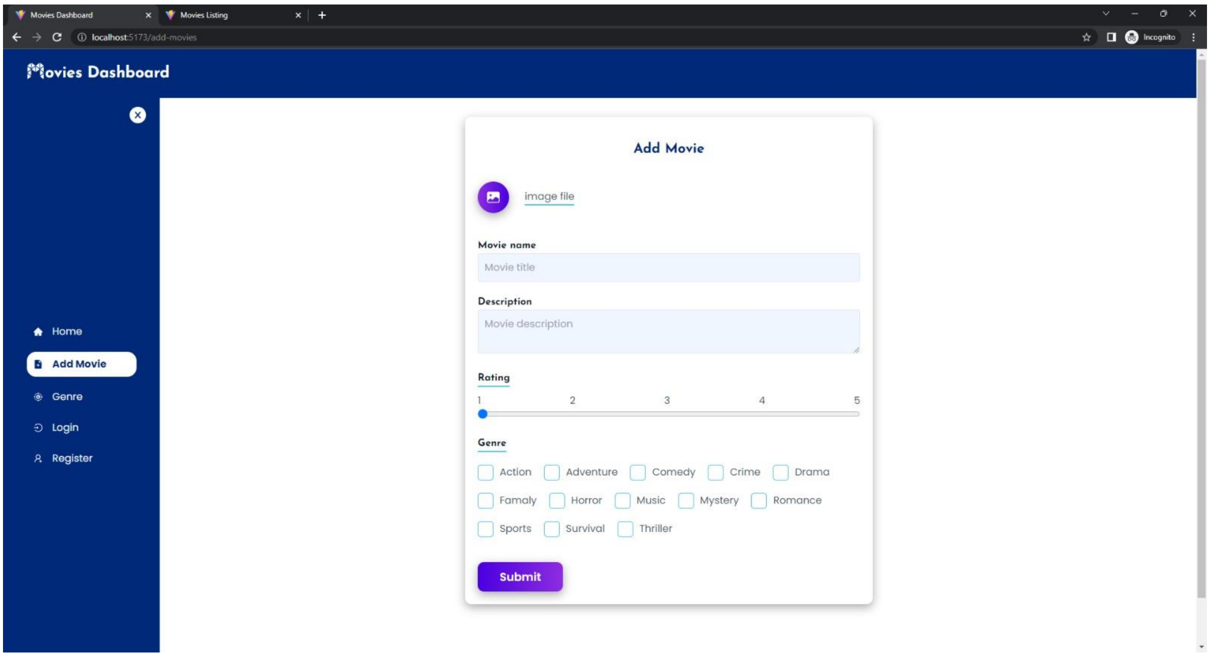
Action Adventure Comedy Crime

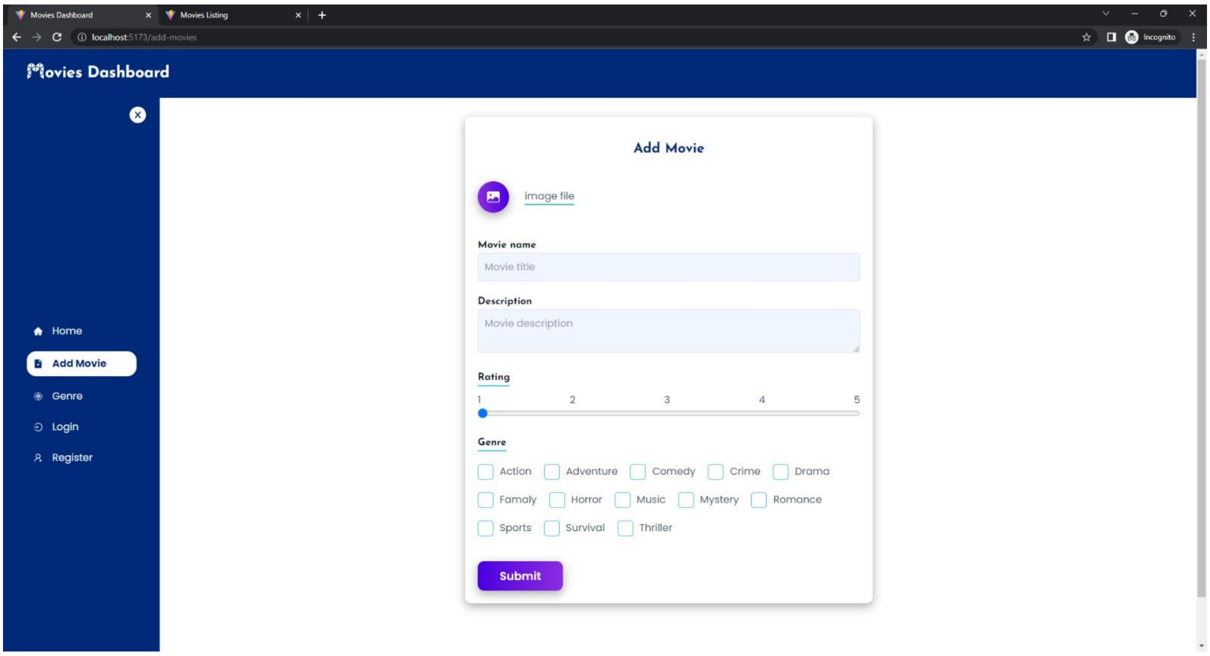
Drama Famaly Horror Music

Mystery Romance Sports Survival

Thriller


Movies Dashboard
Movies Listing
localhost:5173/genre
Incognito
Movies Dashboard
Home
Add Movie
Genre
Login
Register
Edit/Add Genre
Title
Enter Genre
Add
Action
Adventure
Comedy
Crime
Drama
Famaly
Horror
Music
Mystery
Romance
Sports
Survival
Thriller


Movies Dashboard
Movies Listing
localhost:5173/add-movies
Incognito
Movies Dashboard
Home
Add Movie
Genre
Login
Register
Add Movie
image file
Movie name
Movie title
Description
Movie description
Rating
1
2
3
4
5
Genre
□ Action
□ Adventure
□ Comedy
□ Crime
□ Drama
□ Famaly
□ Horror
□ Music
□ Mystery
□ Romance
□ Sports
□ Survival
□ Thriller
Submit



8. ADVANTAGES & DISADVANTAGES

a. Advantages

- **User-Friendly Interface**

I movies offers a visually engaging and intuitive interface where users can easily browse movies, watch trailers, and book tickets.

- **Secure Authentication System**

JWT-based authentication ensures secure access, with role-based differentiation between normal users and admins.

- **Movie Categorization & Search**

Users can explore movies based on genre, language, release year, and use the search feature for quick discovery.

- **Watchlist & Booking System**

Users can add movies to their watchlist and book tickets in a seamless manner, mimicking real-world movie booking platforms.

- **Responsive Design**

The UI is fully responsive, delivering a smooth experience across mobiles, tablets, and desktops.

- **Scalable MERN Stack**

Built using MongoDB, Express, React, and Node.js, I movies is modular, easily maintainable, and ready for future upgrades.

- **Trailer Playback Support**

Integration of trailers through embedded video players enhances user engagement before ticket booking.

b. Disadvantages

- **No Real Payment Integration**
  The ticket booking system lacks integration with live payment gateways (e.g., Razorpay or PayPal).
- **Basic Admin Panel**
  Current admin functionalities are limited to managing movies and users; analytics and logs are missing.
- **No Real-Time Seat Selection**
  Users can book tickets, but real-time seat selection and availability updates are not implemented yet.
- **Limited Role Hierarchy**
  Only user and admin roles are defined; roles like cinema staff or content moderators are yet to be added.
- **No Push Notifications**
  There are no live alerts or push notifications for booking confirmations or movie releases.

---

9. CONCLUSION

The development of **I movies**, a movie browsing and booking platform using the MERN stack, successfully brings together essential features of a modern entertainment application. From browsing movie listings to managing bookings, the application provides a well-rounded user experience backed by secure authentication and responsive design.

This project allowed us to apply full-stack web development concepts including API creation, database schema modeling, session handling, and frontend design. By using MongoDB, Express.js, React, and Node.js, we ensured a fast, scalable, and structured system.

Though currently limited in a few areas like payment processing and analytics, I movies lays the groundwork for a feature-rich and scalable production-level system. The learning from this project included RESTful API implementation, route protection, UI design, and deployment strategies.

Overall, I movies serves as a strong demonstration of our capability to develop robust and dynamic web applications from scratch.

---

10. FUTURE SCOPE

While the current version of I movies offers a core movie booking experience, several enhancements can be implemented in future versions:

a. Payment Gateway Integration

- Integrate real-time payment gateways (Razorpay, Stripe) to enable secure ticket booking.
- Offer UPI, credit/debit card, and wallet payment options.

b. Mobile App Development

- Build a mobile app using React Native to enhance accessibility and provide notifications.
- Use push alerts for movie release updates and booking confirmations.

c. Real-Time Seat Selection

- Implement socket-based live seat selection and updates using WebSockets or Socket.io.

d. Recommendation Engine

- Use ML algorithms to suggest movies based on user history, genre preference, and ratings.

e. Review & Rating System

- Enable users to rate and review movies post-booking for better community feedback.

f. Admin Analytics Panel

- Include data visualizations for user traffic, movie popularity, booking trends using Chart.js/Recharts.

g. Multi-Language Support

- Add support for regional languages and internationalization (i18n) for wider audience reach.

h. Enhanced Security

- Implement CAPTCHA, Two-Factor Authentication (2FA), and session timeout for stronger user security.

i. Ticket PDF Generation

- Provide downloadable e-tickets with QR codes for easy scanning at theaters.

j. Performance & Accessibility

- Optimize image loading, use CDN, and improve accessibility to make the app inclusive and faster.

11. APPENDIX

a. Source Code

Click here to visit my GitHub repository

Find the source code here.

b. Demo Link

Click here to watch the Demo Video

c. Dataset

No external dataset was provided or required in the cloned repository. These files were used to simulate backend responses for development and testing purposes.
