Final Report
iMovies – Online Movie Ticket Booking System

1. INTRODUCTION

1.1 Project Overview

The iMovies project is a full-stack MERN (MongoDB, ExpressJS, ReactJS, NodeJS) application designed to provide users with a seamless movie ticket booking experience. It allows users to explore, book, and manage movie tickets online, while offering an admin panel to manage theaters, shows, and bookings.

1.2 Purpose

The project aims to digitalize the movie ticket booking process, enhancing user experience by providing features like movie discovery, booking history, favorites, and personalized recommendations based on user interests.

2. IDEATION PHASE

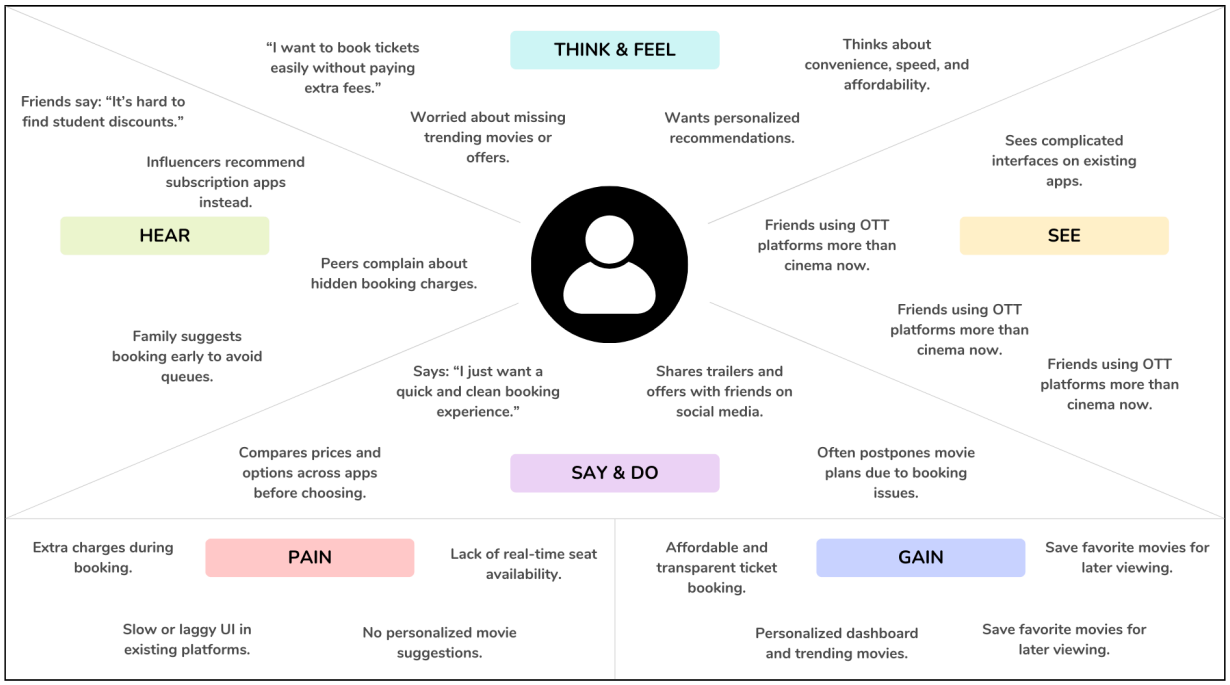
2.1 Problem Statement

Customers often face difficulties in checking real-time seat availability, comparing shows across theaters, and managing bookings efficiently. Traditional systems lack user-specific features and seamless online interaction.

2.2 Empathy Map Canvas

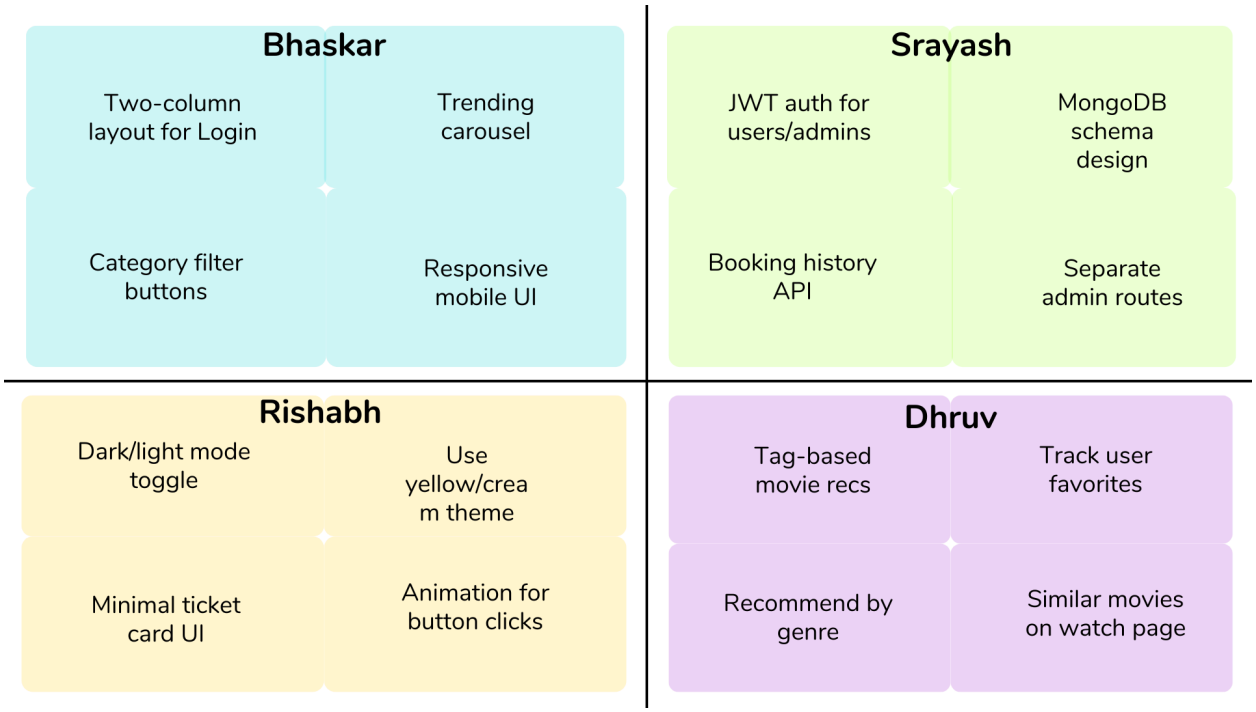
The empathy map helped our team understand user behavior and needs:

- **Think & Feel**: Wants a smooth, fast booking experience.
- **See**: Complicated interfaces, unavailability of live updates.
- **Hear**: Friends complain about bad booking experiences.
- **Say & Do**: Expresses frustration with slow platforms.
- **Pain**: Booking failures, unclear seat availability.
- **Gain**: Wants a reliable, smooth, and intuitive booking process.



2.3 Brainstorming

Each team member suggested ideas ranging from authentication security to personalized movie recommendations, and we prioritized them using an idea priority matrix.



3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

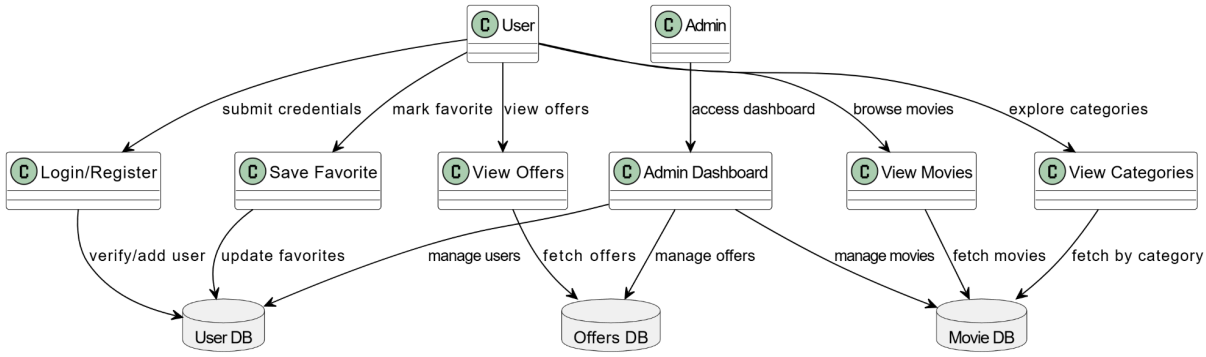
- **Awareness**: Learn about iMovies through social media or peers.
- **Consideration**: Browses movie lists and show timings.
- **Booking**: Selects seats and makes payment.
- **Experience**: Receives ticket and watches movie.
- **Feedback**: Rates or adds movies to favorites.

3.2 Solution Requirement

- **User Side**: Browse movies, book tickets, view history, mark favorites.
- **Admin Side**: Manage theaters, shows, bookings.
- **System**: Secure authentication, optimized performance.

3.3 Data Flow Diagram

A DFD was created to illustrate data interaction between users, backend services, and database modules.



3.4 Technology Stack

- **Frontend**: ReactJS
- **Backend**: NodeJS, ExpressJS
- **Database**: MongoDB

- **Security**: JWT, Bcrypt
- **HTTP Client**: Axios

4. PROJECT DESIGN

4.1 Problem Solution Fit

The solution fits users' needs by solving core issues—providing live seat availability, secure bookings, and a user-friendly interface.

Customer Segment(s) Movie lovers, cinema-goers, college students, working professionals, families who book movie tickets online.	**Customer Limitations** Budget-conscious users, basic smartphones, average internet connectivity, non-tech-savvy people.	**Available Solutions** BookMyShow, Paytm (Popular but charge fees, not personalized). Local apps (limited reach, outdated UI).
Problems / Pains Difficult UI in existing platforms, extra booking charges, lack of personalization, limited local theater support.	**Problem Root / Cause** Centralized platforms profit-driven, not customer-first. Lack hyperlocal theater integration.	**Behavior** Frequently search for showtimes, compare prices, share plans with friends, use offers/coupons intensely.
Triggers to Act Movie promotions, weekend plans, group bookings, social media buzz, holiday releases. **Emotions** Before: Confusion, frustration After: Satisfaction, ease, excitement	**Your Solution** iMovies – A MERN-based, ad-revenue movie booking platform with no hidden fees, personalized recommendations, and seamless booking.	**Channels of Behavior** Online: Mobile apps, websites, social media. Offline: Posters, word of mouth, cinema counters.

4.2 Proposed Solution

A MERN stack-based web platform with separate flows for users and admins, offering smooth navigation, booking, and management features.

4.3 Solution Architecture

Consists of:

- **Frontend (ReactJS)**: User interface with routing and state management.
- **Backend (NodeJS + ExpressJS)**: Handles logic, authentication, and API services.
- **Database (MongoDB)**: Stores user, movie, theater, and booking data.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week 1: Ideation and Problem Statement Finalization

Week 2: Frontend-Backend Setup

Week 3: Implement Authentication & Booking Flow

Week 4: Admin Features & Styling

Week 5: Testing, Debugging, Final Deployment

6. FUNCTIONAL AND PERFORMANCE TESTING

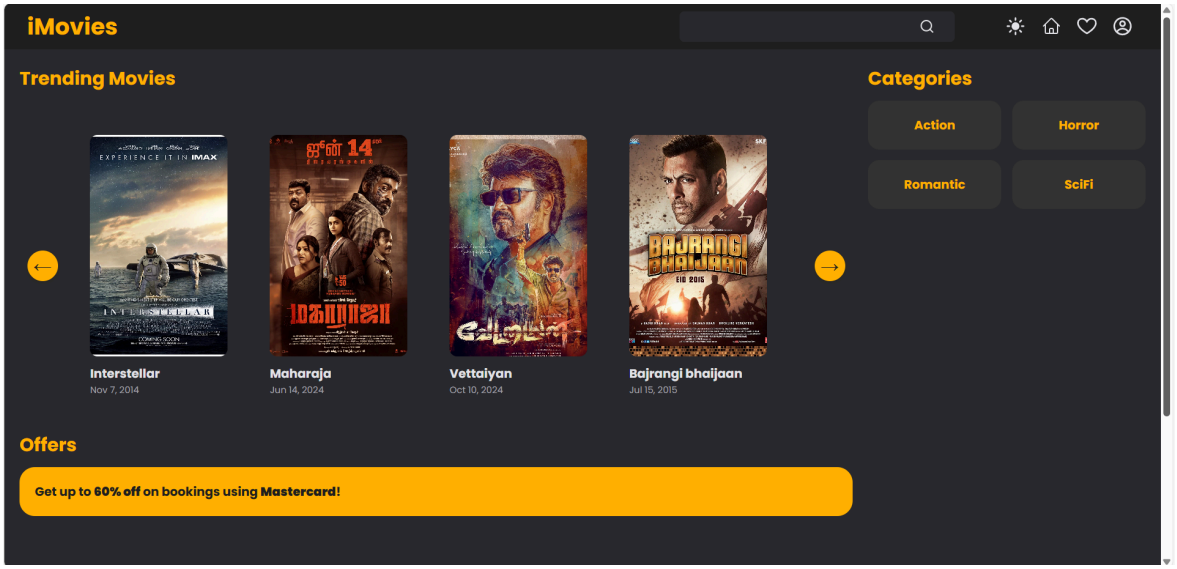
6.1 Performance Testing

- **Tools**: Browser DevTools, Postman
- **Focus**: API response time, login speed, load handling
- **Result**: Stable under load, optimized response time

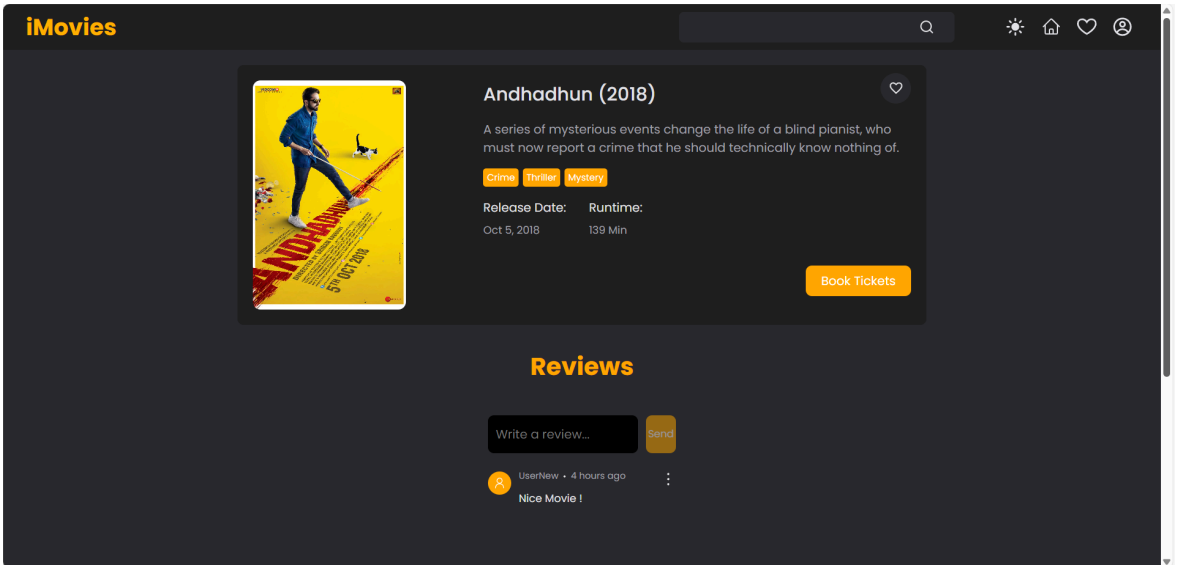
7. RESULTS

7.1 Output Screenshots

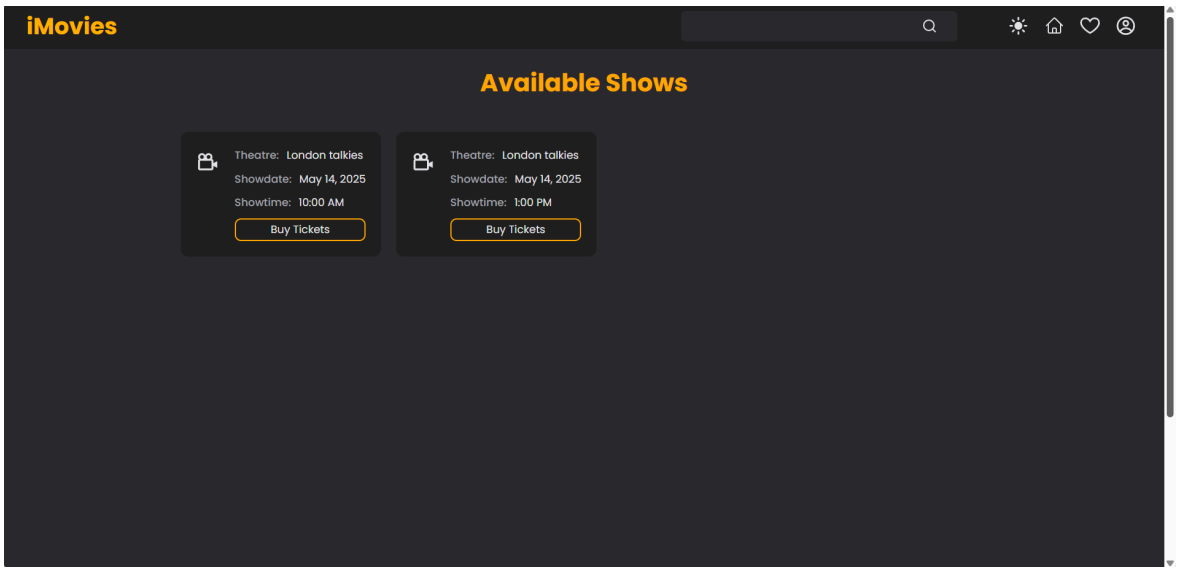
User Homepage



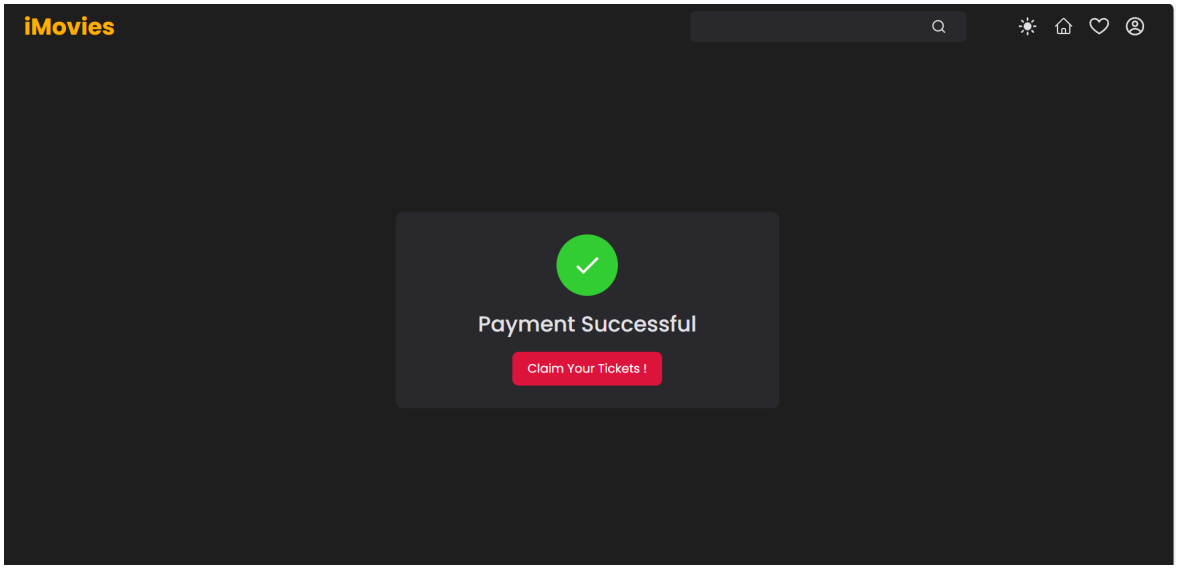
Movie Details Page



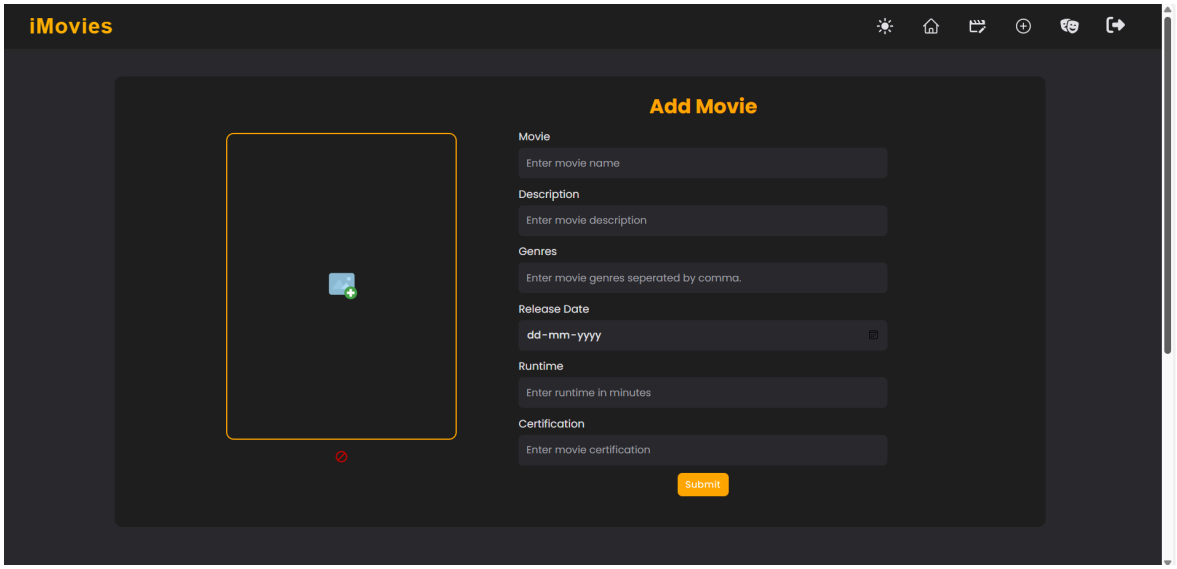
Available Movies Page



Payment Successful Page



Add Movie Page



8. ADVANTAGES & DISADVANTAGES

Advantages:

- Fast and responsive UI
- Real-time booking management
- Personalized recommendations

Disadvantages:

- Not yet deployed on a global scale
- Lacks mobile app version

9. CONCLUSION

iMovies successfully delivers a seamless and efficient ticket booking experience using the MERN stack. With robust features and intuitive design, it addresses common user frustrations and modernizes the booking process.

10. FUTURE SCOPE

- Add mobile app version (React Native)
- Integrate with payment gateways
- Add social login and user reviews
- Improve admin analytics with graphs and reports

11. APPENDIX

Source Code: GitHub Repo

Project Demo Link: Google Drive Demo Video

Team Members:

- Bhaskar Rai – 22BCE11245
- Srayash Kumar – 22BCE11520
- Rishabh Kumar Singh – 22BCE11568
- Dhruv Rai – 22BCG10100