MOVIE TICKET BOOKING

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

Bachelor of Technology

In

Computer Science and Engineering

School of Engineering and Sciences

Submitted by

Somanadh. A	**AP2110010197**
Vignesh. R	**AP2110010176**
Vaibhav. A	**AP2110010131**
Sujith.	**AP22110010342**
Kartikeya	**AP22110010901**

Under the Guidance of

Dr. Ramanjaneyulu,

Assistant professor, Dept.of CSE

SRM University–AP

Neerukonda, Mangalagiri, Guntur

Andhra Pradesh – 522 240

May, 2023

CERTIFICATE

Date: 29-04-2024

This is to certify that the work present in this Project entitled “MOVIE TICKET BOOKING ” has been carried out by Somanadh Vignesh Viabhav Sujith kartikeya under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in School of Engineering and Sciences.

Supervisor

(Signature)

Dr. Ramanjaneyulu

Assistant Professor,

Department of CSE

Acknowledgements

We would like to acknowledge the guidance and support provided by our mentor and advisor Dr. Ramanjaneyulu. Your wisdom and experience helped steer us in the right direction, and your feedback was instrumental in refining our ideas and approaches. I extend my heartfelt thanks to my team members who collaborated tirelessly, sharing their expertise and insights, and working together cohesively towards our common goal. Each of you played a crucial role in shaping the project's outcome, and I am truly grateful for your contributions. Furthermore, I am thankful for the resources and opportunities provided by our university, which enabled us to carry out this project successfully.

ABSTRACT

The Movie Ticket Booking System is a modern web-based application leveraging the MERN stack—MongoDB, Express.js, React.js, and Node.js—to streamline and enhance the experience of booking movie tickets. This platform transforms the traditional reservation process into an efficient digital service, eliminating physical queues and manual seat allocation. Key features of the system include a responsive user interface, real-time seat availability, secure payment integration, and administrative controls for managing theaters and showtimes Additionally, the system focuses on high availability and fault tolerance, supporting a large number of concurrent users. This ensures that during high-traffic events, such as blockbuster openings, the system maintains stability. The project showcases best practices in modern web development, including state management, authentication protocols, and database relationships. By providing both users and administrators with intuitive tools, the system demonstrates a scalable, full-featured solution for the cinema industry.

COMPONENTS :

1. Front-end:
   Technology: **React.js**
2. Back-end:
   Technology: **Node.js** and **Express.js**
3. Database:
   Technology: **MongoDB**

This project includes **AI technologies, open AI's and API's**.

Table of Contents

List of figures

List of tables

1. Introduction

The advent of digital platforms and a rise in internet accessibility, consumer behavior has rapidly shifted towards online services, including entertainment and ticketing. The traditional method of booking movie tickets, often involving long queues, phone calls to box offices, or third-party vendors, no longer meets the expectations of modern users who value speed, convenience, and self-service. This project addresses these challenges by providing a web-based Movie Ticket Booking System designed for simplicity, efficiency, and scalability. It offers a centralized platform where users can register, browse movies based on filters like genre, language, and rating, select their preferred showtime, choose seats in real time, and complete payment through secure gateways. For administrators, the platform provides dashboards to manage theaters, movie schedules, pricing, and customer interactions. By using the MERN stack, the system ensures flexibility and ease of integration with additional features in the future, such as loyalty programs, regional language support, and analytics dashboards.

1.1 Scope

The Movie Ticket Booking System is designed to cater to both customer-facing and administrative functionalities, ensuring a holistic movie theater management and ticketing solution. From a user's perspective, the application offers seamless registration, personalized dashboards, the ability to view currently available and upcoming movies, and the capability to select specific seats during booking. Real-time updates ensure that no two users can book the same seat simultaneously, thus maintaining booking integrity. Furthermore, integration with online payment gateways facilitates secure and instant transactions, reducing dependency on manual payment systems.

From the administrative side, theater managers and staff can access a dedicated admin panel that allows them to manage movie data—such as uploading new movie posters, trailers, and descriptions—set showtimes, configure pricing strategies, and monitor booking trends. The platform is scalable and supports the addition of multiple theaters and cities. In the long run, the system could also integrate third-party APIs for movie metadata, real-time analytics, and digital advertising modules, further extending its commercial applications.

1.3 Definitions, Acronyms, and Abbreviations.

SRS	It Software Requirements Specification – A document that outlines the functional and non-functional requirements of a software system.
OTP	One-Time Password – A password that is valid for only one login session or transaction.
API	Application Programming Interface - A set of rules and definitions that allow software programs to communicate with each other.
UI	User Interface - The space where interactions between humans and machines occur.
DB	Database
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
Razorpay / PayPal	Popular third party online payment gateway services integrated to the systems.
ADMIN	Administrator- The user with the highest level of access, responsible for managing theatre information and user activities.

Table-1

1.4 References

This document and system are developed based on the following references:
1.IEEE SRS Template – IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications.
2. Official Razorpay Developer Documentation: https://razorpay.com/docs

1.5 Overview

This document provides a detailed overview of the software requirements for the Movie Ticket Booking System. It outlines both the functional and non-functional specifications needed for the successful development and deployment of the application.
The system is intended to simplify the process of booking movie tickets for users while allowing theater administrators to manage showtimes, seat availability, and transactions. The system supports real-time seat selection, secure payment processing, and OTP-based authentication. It is designed to be scalable and secure, using cloud-based infrastructure and modern web technologies.
The remainder of this document is structured as follows:
Section 2: Provides an overall description of the system, including its environment, user characteristics, and design constraints.
Section 3: Details the external interface requirements.
Section 4: Covers the system's features and functionalities.
Section 5: Describes non-functional requirements such as performance, security, and maintainability.
Section 6 and Beyond: Contains appendices, business rules, testing strategies, and additional documentation relevant to the system's operation and development.

2. The Overall Description

2.1 Product Perspective

The movie ticket booking system is designed as a comprehensive web-based solution that enables users to book movie tickets online with ease. It simplifies the movie selection, seat booking, and payment process, all through an intuitive and responsive interface. The system functions as an intermediary between moviegoers and theaters, facilitating the booking process while offering various features to ensure a seamless experience.

Context: The system operates within a broader entertainment ecosystem that includes theaters, movie-goers, and third-party services like payment gateways. It aims to integrate with existing theater operations and support future scalability by allowing theaters to manage showtimes, prices, and seat availability dynamically.

Product Integration: The system integrates several technologies:

- **Frontend**: Built with **React.js** to ensure a smooth, responsive UI.
- **Backend**: Powered by **Node.js** and **Express.js**, providing robust API services.
- **Database**: Uses **MongoDB** to store and manage data such as movie listings, bookings, and user details.

- **Payment Gateway**: Integrated with **Razorpay** and **PayPal** for secure online transactions.
- **Authentication**: Uses **JWT (JSON Web Tokens)** for secure, stateless user authentication.
- **Third-Party Services**: Integration with **Twilio** for OTP-based user authentication and **email/SMS notifications**.

This system enables a fully functional movie ticket booking process that meets the needs of both users and administrators

2.2 Product Functions

The core functionality of the movie ticket booking system includes:

User Registration & Authentication:

Users can register via email or social media accounts (Google/Facebook). They are authenticated using JWT, ensuring secure, stateless sessions. Users can log in, reset passwords, and update account information.

Movie Listings & Search:

The system displays movies currently available in theaters, with information such as genre, rating, and showtimes. Users can filter and search based on movie preferences (e.g., genre, rating, language, or location).

Seat Selection & Availability:

Users can view the available seats for a specific movie, select their preferred seats, and see real-time seat availability. Once selected, the seats are temporarily reserved until the booking process is complete.

Showtime Selection:

For each movie, users can select from available showtimes based on the theater's schedule. The system ensures no overlapping showtimes, and users are notified if a show is fully booked.

Booking Confirmation:

After selecting seats and a showtime, users proceed to payment. Upon successful payment, users receive a booking confirmation via email and SMS with details like movie title, theater, showtime, seat numbers, and booking reference.

Payment Gateway Integration:

Users can pay for their tickets securely through integrated payment gateways like Razorpay or PayPal. The payment process includes support for multiple payment methods (credit/debit cards, wallets).

Admin Panel:

Administrators (theater managers) can manage the movie listings, showtimes, and ticket pricing. The system provides real-time statistics on ticket sales, including daily/weekly/monthly analytics. Admins can also handle cancellations, refunds, and updates to the movie schedule.

Ticket Cancellation & Refunds:

Users can cancel their bookings within a predefined time limit (e.g., 20 minutes before the

showtime). Refunds are processed via the same payment gateway.

Notifications & Alerts:
Users receive notifications for booking confirmations, cancellations, and updates about their selected movies (e.g., showtimes or price changes).

User Profile & History:
Users can maintain their profile, view past bookings, and access history for future reference.

2.3 User Characteristics

TThe system is designed for a variety of users, each with specific characteristics:

1. **End Users (Moviegoers)**:
    - **Technical Proficiency**: Users may range from tech-savvy individuals to those with minimal technical knowledge. The system should be intuitive and easy to use for all.
    - **Age Group**: Users of all ages, from young adults to older generations, will interact with the system.
    - **Needs**: Moviegoers need a fast, secure, and easy way to book tickets without the need to visit the theater physically.
2. **Theater Administrators**:
    - **Technical Proficiency**: Admins need basic to moderate technical skills to manage and configure the system, including handling the movie listings, showtimes, and ticket prices.
    - **Needs**: Admins need to manage show schedules efficiently, track booking statistics, and offer real-time customer support when necessary.
3. **System Users (Payment Providers)**:
    - **Technical Proficiency**: Payment providers (Razorpay/PayPal) must provide easy-to-integrate APIs that enable seamless payment processing.
    - **Needs**: They require secure payment handling with transparent transaction processing and validation.
4. **Support Staff**:
    - **Technical Proficiency**: Support staff should be familiar with the system's basic functionality, such as canceling tickets or resolving user issues.
    - **Needs**: A straightforward interface to assist users in case of issues related to bookings, cancellations, or refunds.

2.4 Constraints

While the movie ticket booking system is designed to be flexible, there are some inherent constraints:

1. **Bandwidth and Network Dependency**:
   The system requires a stable internet connection for both users and admins. Slow network speeds may lead to poor user experience, especially during payment processing or seat selection.
2. **Payment Gateway Limitations**:
   Integration with payment gateways like Razorpay and PayPal might have region-specific constraints and currency restrictions. Additionally, payment processing fees could impact pricing strategies for theater owners.
3. **Server and Database Load**:
   As the system scales to handle a larger number of users, database performance, and server load could become a challenge. The system must be optimized to handle high traffic during peak hours (e.g., weekends, new movie releases).
4. **Legal Compliance**:
   The system must comply with relevant data protection regulations (e.g., GDPR, PCI-DSS) for secure handling of users' personal and payment information.
5. **Seat Reservation Time**:
   The time available for users to hold seats during the booking process must be limited to prevent double bookings, particularly during high-demand periods.
6. **Language and Regional Customization**:
   Although the system can be deployed in various regions, it may require language or regional customization for theaters operating in different countries or regions.
7. **Dependency on Third-Party Services**:
   The system's reliance on third-party services (e.g., payment gateways, OTP service providers, and email/SMS providers) could lead to issues if these services experience downtime or changes in their terms of service.

2.5 Assumptions and Dependencies

the movie ticket booking system to function smoothly and meet user expectations, several assumptions about its environment and operation must hold true. Additionally, the system has dependencies on various external services and components. Below are the key assumptions and dependencies that must be considered for the system to perform as intended:

1. Internet Access

Assumption: It is assumed that all users (customers) and administrators (theater staff) will have continuous access to the internet to interact with the system. The system is web-based, so it relies on internet connectivity for both the frontend and backend to function.

Impact: Without a stable internet connection, users will not be able to browse movies, select seats, complete payments, or receive booking confirmations. Administrators will not be able to update movie schedules, manage seats, or view real-time booking data. Therefore, the system cannot operate without reliable internet access at both ends.

2. Third-Party Service Availability
Assumption: The system depends on third-party services like Razorpay for payment processing, Twilio for OTP authentication, and external email/SMS providers for notifications (e.g., booking confirmations, reminders). It is assumed that these services will remain available and function without significant downtime.

Impact: Any disruption in these third-party services could cause issues like:

Payment failures if the payment gateway is unavailable.

Authentication issues if the OTP service fails to send messages.

Communication breakdowns if the email/SMS service fails to deliver booking confirmations and alerts.

3. Theater Integration
Assumption: It is assumed that the theaters integrating with this system will provide accurate, up-to-date information about movie schedules, seat availability, pricing, and other critical data. This information must be regularly updated to reflect any changes in real-time.

Impact: If a theater fails to update movie schedules or seat availability regularly, it could lead to discrepancies between what users see on the system and the actual availability. For example:

Users could select seats that are already taken or choose a showtime that has been canceled.

Inaccurate pricing could lead to customers being charged incorrectly or theaters not receiving the correct payment amount.

4. Data Integrity
Assumption: The system assumes that the data provided by the theaters (such as movie listings, showtimes, seat availability, and ticket prices) is accurate and correctly formatted. The accuracy of this data is critical for a seamless user experience.

Impact: If incorrect data is entered into the system, it could lead to a range of issues:

Users could book seats for a movie that does not exist or for showtimes that are incorrect.

Incorrect pricing could cause billing issues, leading to customer dissatisfaction or financial loss.

Overbooked or unavailable seats may result in double-booking, which can damage the system's reputation.

5. Security Compliance

Assumption: The system assumes that users will follow basic security practices, such as using strong passwords, keeping their browsers up to date, and avoiding sharing account details with unauthorized parties. The system also assumes that administrators will use secure authentication mechanisms to prevent unauthorized access to theater management functionalities.

Impact: If users fail to follow security protocols or use weak passwords, their accounts may be

compromised, leading to unauthorized access, fraudulent transactions, or identity theft. Similarly, weak security practices among administrators could allow malicious actors to access sensitive theater or payment data**.**

6. Scalability

Assumption: The system is assumed to be scalable, capable of handling increasing numbers of users and bookings, especially during peak times such as holidays, weekends, and new movie releases. This includes handling:

A large number of concurrent users searching for showtimes, selecting seats, and making payments.

High transaction volumes during popular movie releases or events.

Increased load on the database and server infrastructure.

Impact: If the system is not scalable, it may experience slow response times, errors, or crashes during peak usage periods. A failure to scale properly could negatively impact user satisfaction and reduce trust in the platform**.**

EXTERNAL INTERFACE REQUIREMENTS

In any system, external interfaces define how the system interacts with other systems, hardware, or software components. For the Movie Ticket Booking System, several interfaces are crucial for ensuring proper functionality and seamless communication. Below is a detailed breakdown of the external interface requirements:

3.1.1 *User Interface Requirements*

Create Button:

The user interface (UI) is the primary means through which users interact with the Movie Ticket Booking System. It should be intuitive, responsive, and provide all necessary functionality for both customers and administrators.

Key UI Components:

1. **Customer Dashboard**:
    - **Login/Registration Page**: Users can create accounts, log in via email or social media, and securely access their personalized dashboard.
    - **Movie Listings**: A dynamic list of movies with posters, showtimes, genres, ratings, and languages. Users can filter and search for movies based on different criteria (e.g., genre, language, release date).
    - **Seat Selection**: A visual layout of the theater's seating arrangement that allows users to select available seats in real-time.
    - **Payment Gateway Integration**: A secure and user-friendly interface for entering payment details (credit card, wallet, etc.), including a confirmation screen and failure handling.
    - **Booking History**: Users can view their past bookings, including showtimes, seats, and payment information.
2. **Admin Dashboard**:
    - **Login Page**: Secure login page for administrators to access the management interface.
    - **Movie Management**: Interface to add or remove movies, update showtimes, and manage movie metadata (e.g., posters, trailers).
    - **Theater & ShowTime Management**: Admins can update theater layouts, assign movie schedules, and set pricing.
    - **Booking Monitoring**: Real-time monitoring of bookings, allowing administrators to track the number of available seats, sold tickets, and payment status.
    - **User Management**: Admin interface to manage user accounts, view transaction histories, and handle customer support inquiries.

Design Considerations:

- The UI should be **responsive**, ensuring compatibility across various devices (smartphones, tablets, desktops).
- A **consistent and simple design** for better usability.
- **Accessibility features** (e.g., text resizing, screen reader support) to accommodate a diverse range of users

3.1.2 Hardware Interface Requirements

Key Hardware Components for the Movie Ticket Booking System

The Movie Ticket Booking System, while primarily a web-based application, interacts with various hardware components to deliver a seamless experience to users and administrators. These hardware components ensure the system is scalable, reliable, and efficient, especially when integrated with physical infrastructures in theaters.

Servers and Cloud Infrastructure

The backbone of the system's operation is the server and cloud infrastructure. This hardware infrastructure is responsible for processing, storing, and managing the vast amount of data generated by users interacting with the system, as well as ensuring continuous operation even during periods of high demand.

- **Server Requirements**: The system requires robust servers capable of handling web requests, processing backend logic, interacting with databases, and ensuring smooth communication with external systems like payment gateways and email services. These servers need to be highly available and scalable to support the dynamic nature of the movie ticket booking environment, especially during peak times such as holiday weekends or movie releases.
- **Cloud Platforms**: Hosting the system on cloud platforms such as **AWS (Amazon Web Services)** or **Google Cloud** allows the system to leverage scalable resources. These platforms provide virtual servers, storage, and additional tools like load balancing and auto-scaling, which ensure that the application can handle varying traffic loads effectively. The cloud also offers redundancy, ensuring that if one server fails, another can seamlessly take over, maintaining uninterrupted service.
- **Backend Processing**: The servers are responsible for processing critical tasks like user registration, movie listing retrieval, booking management, and payment transactions. The high availability of these servers is crucial for real-time updates and seamless performance for users.

Point of Sale (POS) Terminals

In physical theaters, users may prefer to collect their tickets in person. In such scenarios, the system must integrate with the theater's **Point of Sale (POS) systems** to verify and issue physical tickets. The POS terminal is responsible for processing the ticket confirmation and generating a printed copy of the ticket, allowing users to enter the theater.

- **Ticket Verification**: The POS systems will verify the booking details stored in the backend, ensuring that the ticket is valid for the specific movie and showtime. This interaction between the Movie Ticket Booking System and the POS terminal enables a smooth transition from online booking to physical ticket issuance, minimizing errors and ensuring accurate transaction records.
- **Integration**: The integration between the web-based booking system and POS terminals must be seamless, ensuring that all details, such as movie name, showtime, seat selection, and payment status, are accurately reflected in the POS terminal. This requires a reliable communication protocol between the system’s backend and POS hardware to avoid discrepancies.

Barcode/QR Code Scanners

For users who have booked their tickets online and chosen to either receive a digital or physical ticket, **barcode/QR code scanners** play a critical role in facilitating ticket validation at the theater entrance. The system will generate a unique QR code for each booking, and this code must be scanned at the theater for entry verification.

- **QR Code Generation**: Upon completing the booking, the system will generate a unique **QR code** that contains essential details such as booking ID, movie name, showtime, and seat information. This QR code serves as a digital "ticket" and can be scanned directly from the user's smartphone or printed ticket.
- **Scanner Requirements**: The hardware at the theater (barcode scanners or mobile scanning devices) must be able to read the QR code quickly and accurately. These scanners are linked to the backend system, which verifies the booking in real-time. If the code matches the booking information, the system will validate the ticket and allow entry.
- **Real-time Data Sync**: Since the QR code scanning process must occur in real-time, the hardware should be equipped with fast processing capabilities to minimize waiting times for users and prevent delays at the theater entrance.

Requirements for Hardware

The Movie Ticket Booking System's performance and reliability are dependent on the availability of certain hardware resources, both for users interacting with the platform and for administrators managing the system.

- **Admin and User Devices**:
    - **Admin Users**: Admins require **computing hardware**, such as desktops or laptops, to access the backend dashboard of the system. These devices will run web browsers to interact with the administrative functions, such as movie management, ticketing reports, and booking monitoring.
    - **Customer Users**: Users will access the system through devices like smartphones, tablets, or laptops. These devices must support modern web browsers to interact with the booking interface, select seats, process payments, and receive notifications. Internet access is essential for users to engage with the system in real time, and they need devices with sufficient processing power to handle media-rich content, such as movie posters and trailers.
- **Backend Systems**: The system's backend, hosted on cloud platforms or physical servers, requires sufficient hardware resources to handle real-time processing of data, large-scale user interactions, and high-volume transactions, especially during peak usage periods. These backend systems must have adequate **CPU**, **RAM**, and **storage** to ensure that the system operates smoothly under varying load conditions. Additionally, databases hosting user data, booking records, and movie details must be optimized for quick data retrieval and transaction processing.

Scalability and Redundancy of Hardware

As the system grows and attracts more users, the demand for resources will increase. The hardware infrastructure must be scalable to accommodate growth, particularly when the system experiences high traffic during peak periods. The servers must be capable of scaling horizontally (adding more servers) or vertically (upgrading existing server resources) to meet the increased demand.

- **Load Balancing**: In cloud environments, the use of **load balancing** ensures that traffic is evenly distributed across servers, preventing any single server from becoming overloaded and ensuring high availability.
- **Redundancy**: The system's hardware must be resilient to failure. Redundancy mechanisms, such as backup servers, multiple data centers, and failover systems, ensure that if one part of the infrastructure fails, others can take over without service disruption.

By integrating these key hardware components and ensuring that they are optimized for performance, the Movie Ticket Booking System can provide a smooth, reliable, and efficient service for users and administrators alike, whether they are interacting through a web browser or relying on physical infrastructure at the theater.

3.1.3 Software Interface Requirements

The Movie Ticket Booking System will interact with various software components and APIs to provide complete functionality. These software interfaces ensure that the system can perform tasks like payment processing, SMS/email notifications, and real-time updates.

Key Software Interfaces:

1. **Payment Gateway Integration**:
    - The system interfaces with payment gateways like **Razorpay**, **Stripe**, or **PayPal** for handling secure transactions. These external services provide APIs for processing payments, handling refunds, and ensuring compliance with payment security standards (e.g., PCI-DSS).
2. **Authentication Systems**:
    - Integration with third-party authentication systems (e.g., **Google Login**, **Facebook Login**) for user registration and login via social media accounts.
3. **Notification Services**:
    - The system uses services like **Twilio** or **SendGrid** for sending **SMS** and **email notifications** for booking confirmations, reminders, and updates to users. These services offer APIs for easy integration with the system.
4. **Cloud Storage Services**:
    - The system may use cloud storage services such as **Amazon S3** or **Cloudinary** to store and retrieve movie posters, trailers, and other media files. These services provide APIs to upload, transform, and serve media content efficiently.
5. **Database Interfaces**:
    - The system uses **MongoDB** and **Mongoose** to interface with the NoSQL database for storing user information, movie listings, and bookings. The system will communicate with the database using APIs for CRUD (Create, Read, Update, Delete) operations.
6. **Content Management Systems (CMS)**:
    - Administrators may use a third-party CMS for managing movie descriptions, trailers, or images. The system will integrate with the CMS through REST APIs.

Software Requirements:

- RESTful APIs for communication between frontend and backend.
- Security standards for payment and user authentication (e.g., **JWT** for secure login sessions).
- Compatibility with modern web browsers (e.g., **Chrome**, **Firefox**, **Edge**).

3.1.4 Communication Interface Requirements

The system must support several communication protocols to ensure seamless data exchange between the client-side (frontend), backend services, and external systems.

Key Communication Interfaces:

1. **HTTP/HTTPS**:
    - The system will use **HTTP/HTTPS** protocols for communication between the client (user's web browser) and the server. **HTTPS** will ensure secure communication, especially during payment processing and user login.
2. **RESTful APIs**:
    - The system will expose RESTful APIs for handling requests related to movie listings, seat bookings, and user management. These APIs will be used by the frontend to fetch data and perform operations in real-time.
3. **WebSockets (Optional)**:
    - For real-time features like seat availability updates and notifications, the system could use **WebSockets** or similar technologies. This allows for bidirectional communication between the server and the client to reflect changes (e.g., seat selection updates) immediately without requiring page reloads.
4. **SMS and Email Protocols**:
    - The system will use **SMS** and **email** protocols to notify users about booking confirmations, reminders, and updates. The communication with services like **Twilio** and **SendGrid** will be done using their respective APIs.
5. **Payment Gateway Communication**:
    - The system will communicate with the payment gateways via secure API calls using **HTTPS**. The communication will include transaction details, user information, and payment confirmations.
6. **Authentication and Session Management**:
    - The system will use secure authentication protocols like **OAuth 2.0** for social logins and **JWT (JSON Web Tokens)** for managing user sessions, ensuring secure user authentication and session management during interactions.

Communication Requirements:

- Secure communication channels (e.g., HTTPS) for sensitive data like payment details.
- Real-time communication support for seamless seat availability and user experience updates.
- Integration with third-party notification services for SMS and email communication.

4.System Features

The Movie Ticket Booking System is designed to provide a seamless, user-friendly, and efficient platform for both customers and theater administrators. Below are the key features that define the system:

4.1 User Registration and Authentication

- **Feature Description**: Users can register on the platform using their email address or social login (e.g., Google). The system supports OTP-based email verification for new user registration and login.
- **Functionality**:
    - New users can create an account with basic details (name, email, password).
    - Existing users can log in using email/OTP or social media accounts.
    - Password recovery options are available through email-based recovery links.

4.2 Movie Listings and Showtimes

- **Feature Description**: The system displays a comprehensive list of movies currently showing, along with their schedules, theaters, and seat availability.
- **Functionality**:
    - Users can browse through a list of available movies, with filters to sort by genre, language, or popularity.
    - Movie showtimes are listed for each theater, allowing users to select their preferred showtime.
    - Movie details, including plot summaries, cast, and posters, are available to users for informed decision-making.

4.3 Seat Selection

- **Feature Description**: The system enables users to select their preferred seats from an interactive seat map.
- **Functionality**:
    - Users can view available, occupied, and reserved seats in real-time.
    - The seat map allows users to select specific seats or choose the "best available" option for their selected showtime.
    - Users can modify their seat selection at any time before payment.

4.4 Payment Gateway Integration

- **Feature Description**: The system integrates with payment gateways such as **Razorpay** and **PayPal** to process payments securely.
- **Functionality**:
    - Users can pay for their bookings using credit/debit cards, wallets, or other supported methods.
    - Multiple currency support allows users to make payments in different regions.
    - The system supports partial payment for bookings that can be adjusted later or canceled within the specified window.

4.5 Ticket Confirmation and Notifications

- **Feature Description**: Once payment is completed, users receive an instant ticket confirmation with a unique QR code for entry.
- **Functionality**:
    - The system sends a confirmation email and SMS containing the ticket details (showtime, seat number, theater, etc.).
    - Users receive a unique QR code for entry verification at the theater.
    - Notifications are also sent for booking reminders or cancellations.

4.6 Booking History and Management

- **Feature Description**: Users can view their past bookings, including canceled or modified tickets.
- **Functionality**:
    - A history page shows a list of previous movie bookings along with details like movie name, showtime, and status.
    - Users can cancel or modify bookings within the allowed time frame (typically 20 minutes before the showtime).

4.7 Admin Dashboard

- **Feature Description**: The system includes an administrative interface that allows theater owners and admins to manage their operations, such as movie listings, showtimes, bookings, and payments.
- **Functionality**:
    - Admins can add or remove movies from the system, update showtimes, and set ticket prices.
    - Admins can view booking reports, user activity, and payment statuses.
    - The dashboard provides real-time analytics on seat occupancy, revenue, and customer feedback.

4.8 Movie Recommendations

- **Feature Description**: The system offers personalized movie recommendations based on user preferences and booking history.
- **Functionality**:
    - The system analyzes user interactions and provides tailored movie suggestions based on genres, previously watched movies, or user ratings.
    - Recommendations are displayed on the homepage or during the booking process.

4.9 User Reviews and Ratings

- **Feature Description**: After watching a movie, users can rate the movie and leave feedback based on their experience.
- **Functionality**:
    - Users can rate movies on a scale (e.g., 1 to 5 stars) and leave text reviews.
    - Reviews and ratings are displayed on the movie detail pages, helping other users make informed decisions.

4.10 Social Media Integration

- **Feature Description**: The system supports sharing booking details on social media platforms.
- **Functionality**:
    - After confirming a ticket booking, users can share their experience on platforms like Facebook, Twitter, or Instagram.
    - Social media sharing includes details such as the movie name, showtime, and theater, encouraging others to join the booking.

4.11 Multilingual and Regional Support

- **Feature Description**: The system provides language and regional customization for theaters in different locations.
- **Functionality**:
    - The user interface can be displayed in multiple languages, based on user preferences or regional settings.
    - The system supports localization of movie listings, prices, and showtimes for different regions or countries.

4.12 Ticket Cancellation and Refund

- **Feature Description**: Users can cancel their bookings within a specified time window and request refunds.
- **Functionality**:
    - Users can cancel their tickets within 20 minutes of booking or before the showtime, depending on theater policies.
    - Refunds are processed through the same payment gateway used during the booking, and users are notified once the refund is issued.

4.13 Theater and Movie Search

- **Feature Description**: The system allows users to search for specific movies or theaters.
- **Functionality**:
    - Users can search by movie title, theater location, or specific showtimes.
    - Advanced filters allow users to narrow down results based on distance, price range, or ratings.

5. Other Nonfunctional Requirements

Non-functional requirements are essential for ensuring the overall quality, scalability, and stability of the Movie Ticket Booking System. These requirements cover performance, software system attributes, and business rules that are critical for both users and administrators.

Performance Requirements

Performance requirements describe the system's ability to handle high loads and ensure responsiveness.

5.1.1 Capacity

The Movie Ticket Booking System must support a high number of users interacting simultaneously, especially during periods of increased demand such as weekends, holidays, or blockbuster movie releases. This includes:

Supporting a minimum of **10,000 concurrent users** performing different operations such as browsing movies, checking seat availability, or booking tickets.

Handling multiple simultaneous transactions without slowing down or crashing.

Managing large volumes of data including user accounts, movie listings, showtimes, seat bookings, and transaction history.

5.1.2 Dynamic Requirements

The system should dynamically adapt to usage spikes. This includes:

Auto-scaling capability to increase server resources when traffic increases and decrease them when traffic drops, thereby optimizing cost and performance.

Load balancing mechanisms to distribute requests evenly across multiple servers to prevent any single point of failure.

Responsive UI components that adjust performance and display based on device specifications and screen sizes.

5.1.3 Quality

To ensure that users have a seamless experience, the system must be of high quality in terms of performance, interface, and response:

Fast page loading and seamless navigation.

Real-time updates in seat selection to avoid double booking or seat conflicts.

High accuracy in ticket generation, confirmation, and record storage.

Reliable and secure payment process that completes within **3-5 seconds** under optimal conditions.

5.2 Software System Attributes

These attributes define how the system should behave in terms of reliability, availability, security, and maintainability.

5.2.1 Reliability

The system must function correctly over time without crashing or data loss:

The system must remain stable even under unexpected conditions.

Frequent tasks like booking, canceling, or updating must execute without errors.

Backup mechanisms must exist to restore lost data in case of system failure.

System components should be tested extensively to prevent runtime issues.

5.2.2 Availability

Availability refers to how accessible the system is to users:

The platform should be accessible **24/7** to users and theater admins.

Scheduled maintenance windows must be communicated in advance.
Failover systems and redundancy must be implemented so that if one server fails, another can take over seamlessly.

5.2.3 Security

Security is critical, especially when dealing with personal and financial information:
End-to-end encryption must be used for all transactions and personal data.
Role-based access control (RBAC) must ensure that only authorized users can access sensitive admin features.
Token-based authentication (JWT) and **multi-factor authentication (MFA)** must be used for secure logins.
User passwords must be stored using **strong hashing algorithms**, and sensitive information like credit card data must comply with **PCI-DSS** standards.
All inputs must be validated and sanitized to prevent security threats like **SQL injection**, **XSS**, and **CSRF attacks**.

5.2.4 Maintainability

The system must be designed in a modular and organized way so it can be maintained and upgraded easily:
Source code should follow **standard naming conventions**, be well-commented, and follow best coding practices.
The system should support **continuous integration/continuous deployment (CI/CD)** for faster updates and debugging.
System logs should help track down bugs or performance issues for faster troubleshooting.
Documentation must be thorough to allow future developers to understand and enhance the system quickly.

Business Rules

T Business rules are the guidelines and operational policies that govern how the system should function in a real-world business context.

User Account and Authentication:

Every user must create a unique account with a valid email ID and phone number.
Users must verify their email/phone using OTP before booking tickets.
A single email or phone number cannot be associated with multiple accounts.
After three unsuccessful login attempts, the user account will be temporarily locked for **10 minutes** to prevent brute-force attacks.

Seat Booking and Reservation Policy:

When a user selects a seat, it is temporarily locked and reserved for **15 minutes** to complete the payment.
If the payment is not completed within that window, the seat is released automatically for other users.
Users cannot reserve more than a predefined number of seats (e.g., 10) per transaction to prevent hoarding.

Payment and Refund Rules:

Full refunds are allowed if the booking is canceled within **20 minutes** of payment, except in cases where the showtime is less than 1 hour away.
Partial refunds or no refunds apply after the 20-minute window, depending on theater policies.
If the movie or show is canceled by the theater, users are entitled to a **full refund** or the option to reschedule.
Refunds are initiated to the original payment method and processed within **5–7 business days**.

Theater Management Responsibility:

Theaters are responsible for uploading accurate information regarding movies, showtimes, pricing, and seat availability.
Admins must ensure real-time synchronization of data between the backend system and the front-end UI.
QR/Barcode Usage:
Every successful booking generates a unique **QR code or barcode**, which users must present at the theater for verification.
Duplicate or expired QR codes will not be accepted at the entry gate.
Promotions and Discounts:
Discounts, promotional codes, and loyalty rewards are subject to conditions like validity period, limited usage, or specific user types (e.g., students, senior citizens).
Promotions cannot be combined unless explicitly mentioned.
Content and Regional Restrictions:
Age-restricted content (e.g., "A" certified movies) will only be shown to users who meet the age criteria and provide valid proof at entry if required.
Language and region-specific listings will be shown based on the user's selected preferences or detected location.

6. Other Requirements

In addition to the functional and non-functional requirements already described, the system has several other important requirements that ensure proper compliance, documentation, and future scalability.

6.1 Legal and Regulatory Compliance

The system must adhere to legal and regulatory standards related to:

- **Data Protection Laws** (such as GDPR, CCPA): Ensuring the protection of users' personal data.
- **PCI-DSS Compliance**: For secure payment processing.
- **Digital Communication Regulations**: Compliance with SMS/email notification laws to avoid spamming users without consent.

6.2 Accessibility Requirements

The system should be usable by people with disabilities by complying with:

- **WCAG 2.1 (Web Content Accessibility Guidelines)** to provide support for screen readers and keyboard navigation.
- Use of **alt text**, **semantic HTML**, and high-contrast UI modes for visually impaired users.

6.3 Localization and Internationalization

To support users from different regions, the system should:

- Be capable of rendering UI in **multiple languages**.

- Support **various currencies** and **regional payment methods**.
- Adjust formats (dates, times) according to regional standards.

6.4 Documentation

The system must include:

- **User manuals** and **FAQs** to help end-users.
- **Developer documentation** for APIs, backend logic, and deployment instructions.
- **Change logs** and **version histories** for future maintenance and upgrades.

6.5 Environmental Requirements

The system should be environmentally friendly by:

- Supporting **energy-efficient hosting solutions**.
- Reducing the need for physical ticket printing through **digital ticketing**.

Appendix A: Glossary

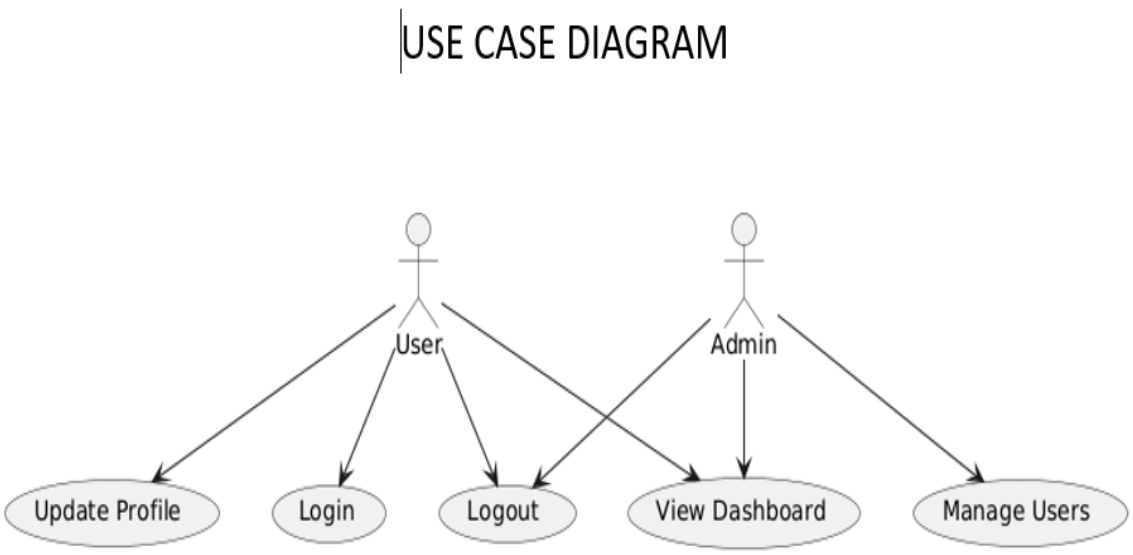
This section defines key terms used throughout the documentation of the Movie Ticket Booking System.

Term	Definition
User	A customer who uses the platform to browse movies, book tickets, and make payments.
Admin	An authorized user (usually theater staff) who manages movie listings, timings, pricing, and bookings.
Showtime	A scheduled time for a particular movie screening.
Seat Map	A graphical representation of available and booked seats in a theater hall.
Payment Gateway	A third-party service (e.g., Razorpay, PayPal) that handles online payments securely.
OTP (One-Time Password)	A temporary code sent via SMS or email for user verification and login.
QR Code	A scannable code generated after booking, used for entry at the theater.
POS (Point of Sale)	A system used at the theater counter for physical ticket management and validation.
GDPR	General Data Protection Regulation — EU law on data protection and privacy.
PCI-DSS	Payment Card Industry Data Security Standard for securing credit card transactions.
UI (User Interface)	The visual part of the system that users interact with.
API (Application Programming Interface)	A set of rules that allows different software systems to communicate.
Concurrency	The ability of the system to handle multiple tasks at the same time.
Load Balancing	The process of distributing network traffic across multiple servers.
Scalability	The ability of the system to grow and handle more load with minimal changes.
WCAG	A set of accessibility guidelines for making web applications user-friendly for individuals with disabilities.

6.Diagrams

Use Case Diagram

Figure-1: UseCase Diagra



DFD Level-0

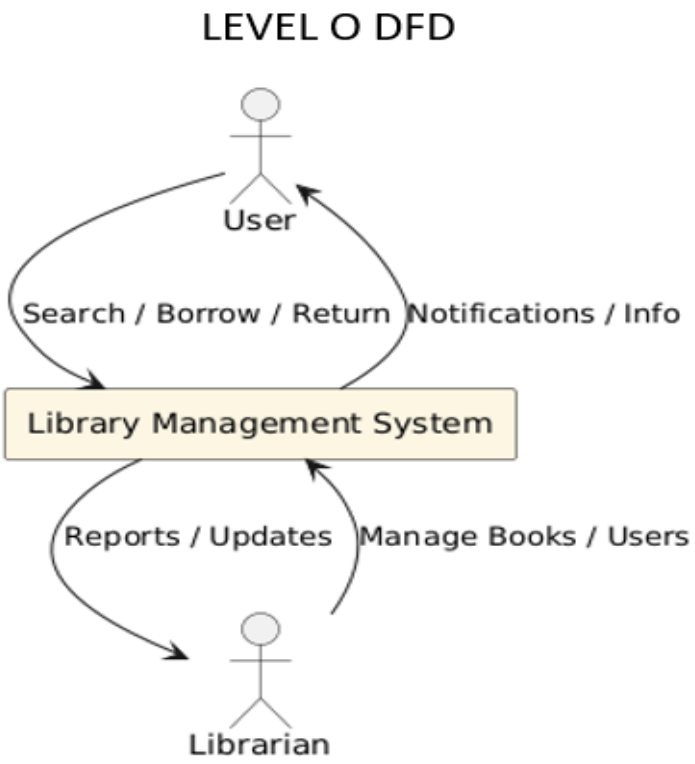
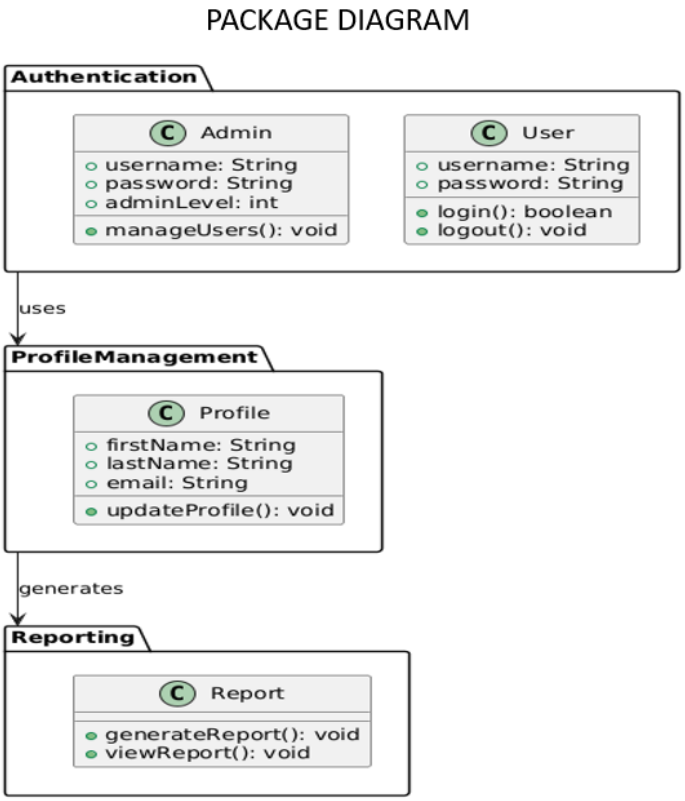


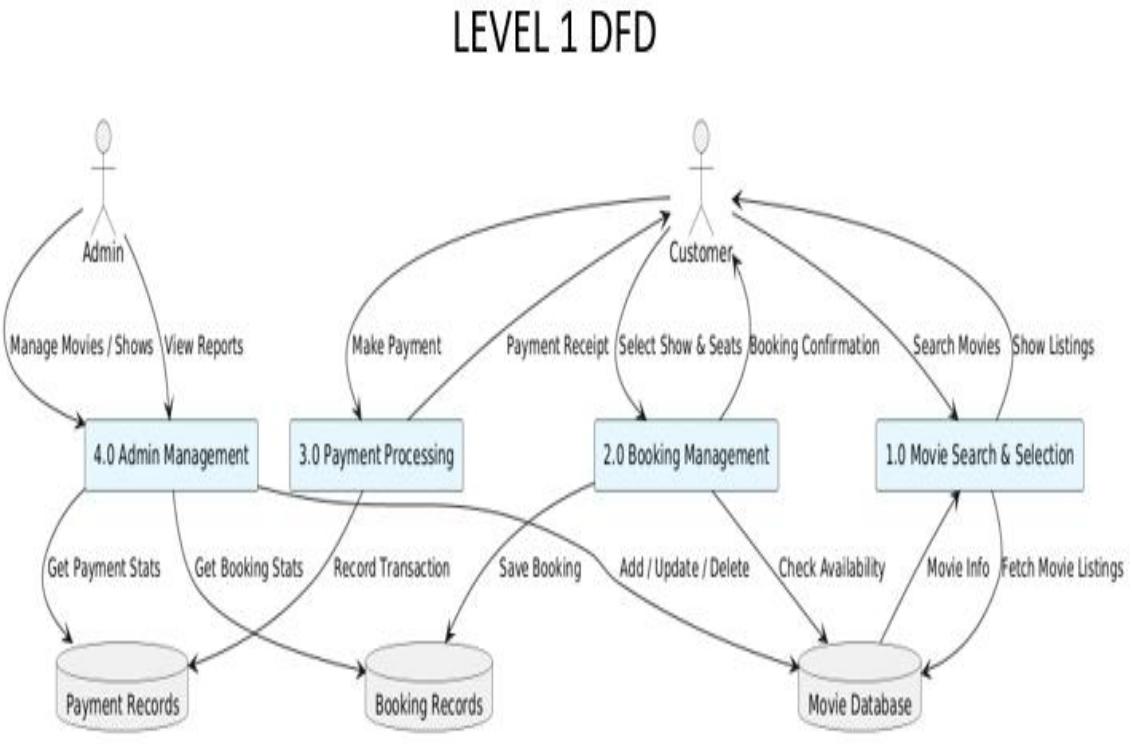
Figure-2: DFD Level-0

PACKAGE DIAGRAM



DFD Level-1

Figure-3: DFD Level-1



DFD DATA DICTIONARY

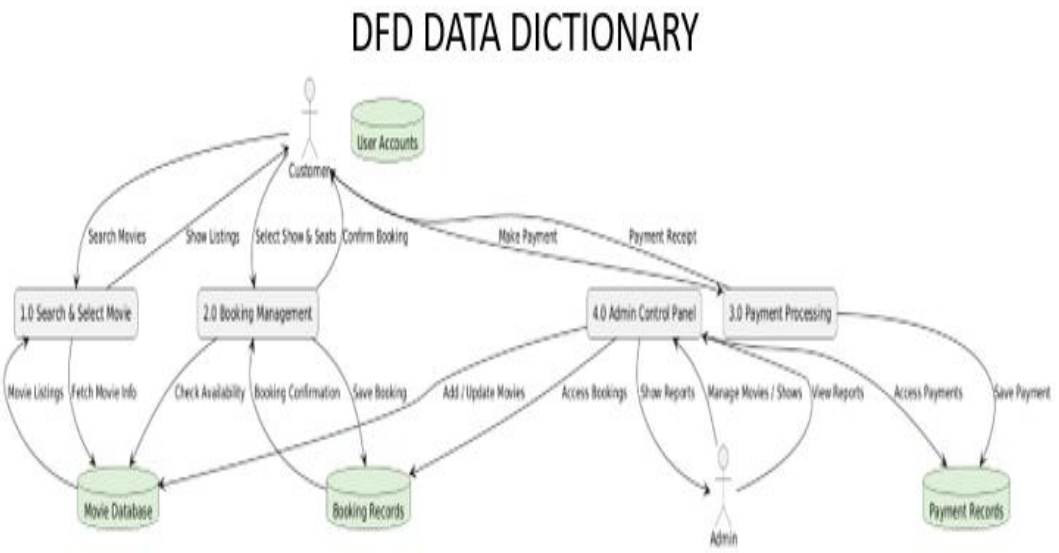
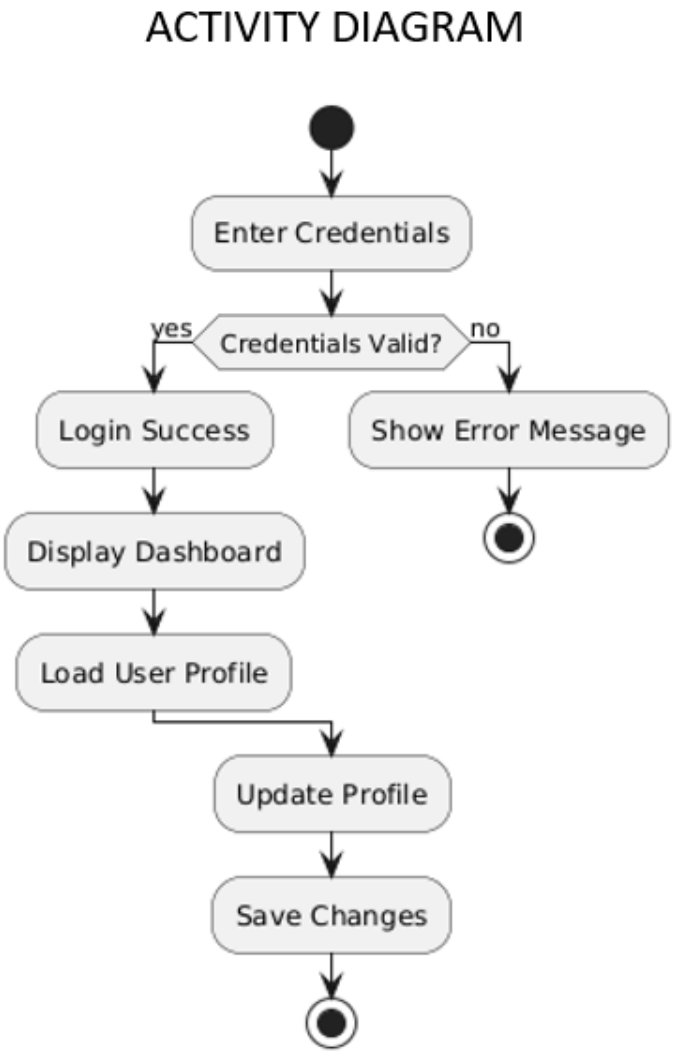


Figure-4: DFD Level-2

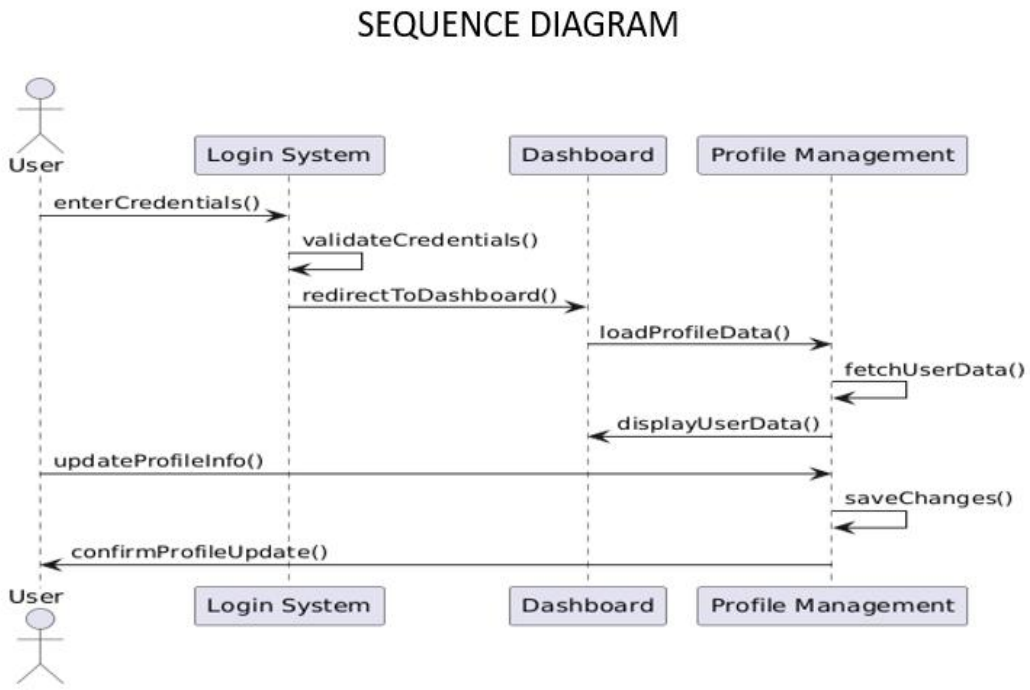
Activity Diagram

Figure-5 : Activity Diagram



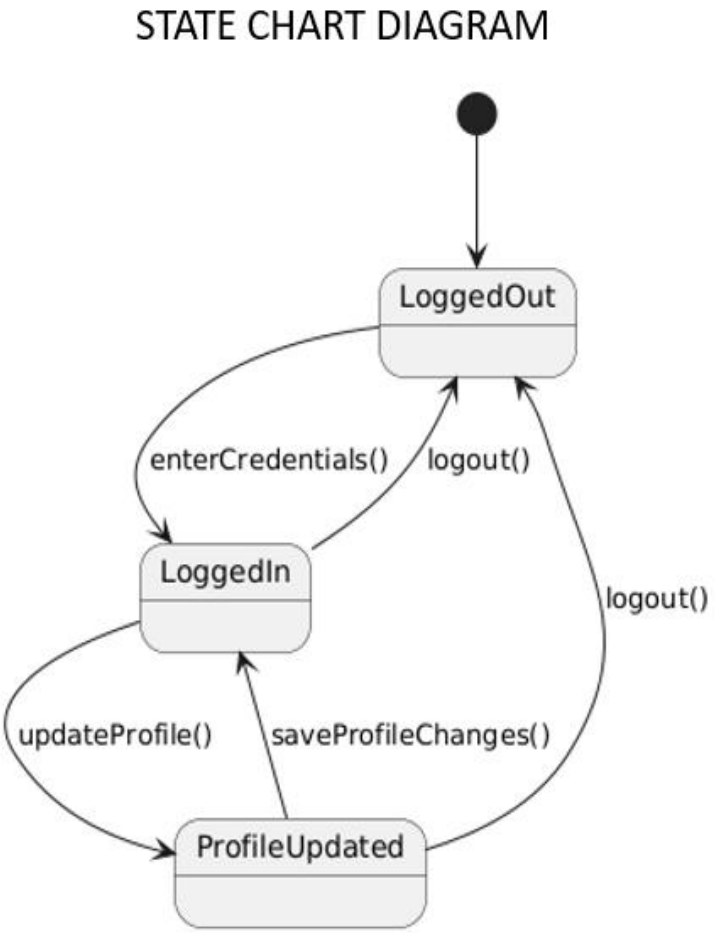
SEQUENCE DIAGRAM

Figure-6: Class Diagram



State Diagram

Figure-7: State Diagram



ER DIAGRAM

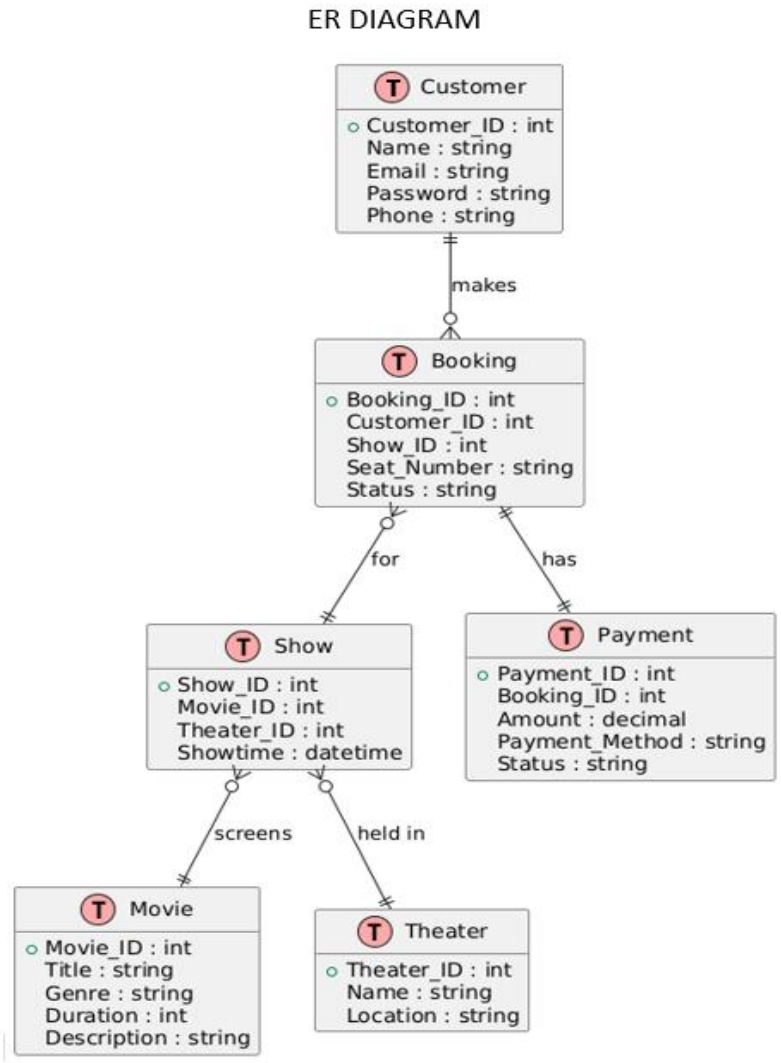


Figure-8: ER Diagram

CLASS DIAGRAM

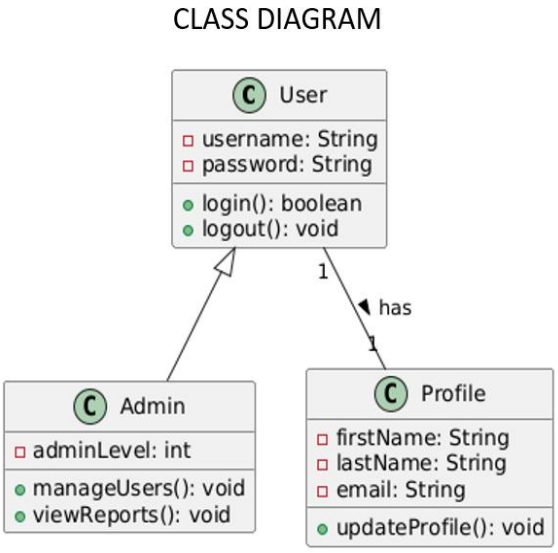
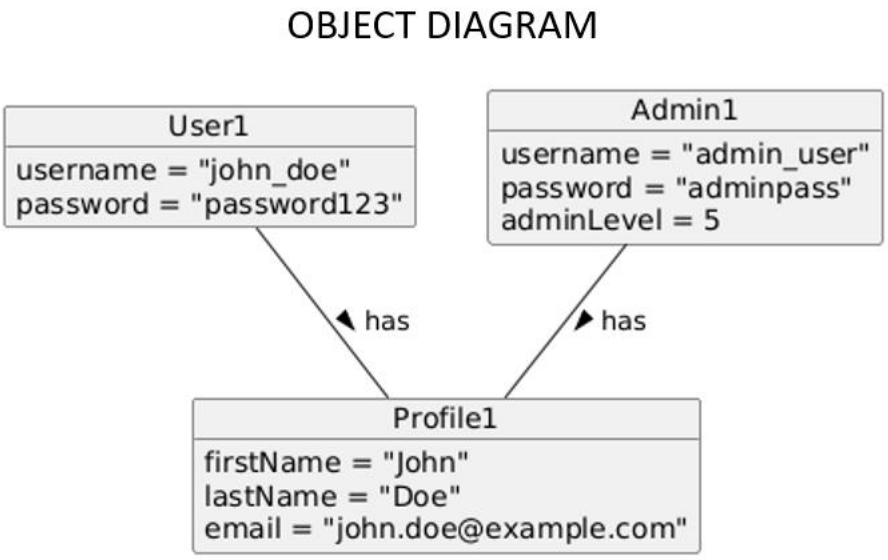


Figure-9: Class Diagram

OBJECT DIAGRAM



7. FRONT END DESCRIPTION

React.js Frontend Overview:

The frontend of this application is built using React.js, a popular JavaScript library known for its flexibility in creating dynamic and interactive user interfaces. React.js facilitates component-based development, allowing for the creation of reusable UI elements.

Card Component:

The Card component represents a visual card within the application's interface. It displays an image along with associated metadata such as the prompt and author's name. Users can interact with the card to download the image.

FormField Component:

The FormField component is responsible for rendering various input fields within forms throughout the application. It provides functionalities for handling user input, such as text entry and checkbox selection, with support for dynamic updates.

Loader Component:

The Loader component offers a visual indication of loading states within the application. It displays an animated spinner, signaling to users that content is being fetched or processed in the background.

React Router DOM:

React Router DOM is utilized for client-side routing in the application. This enables seamless navigation between different views or pages within the application, enhancing the user experience by providing a single-page application feel.

State Management with React Hooks:

State management within components is achieved using React's built-in hooks, primarily useState and useEffect. These hooks enable components to manage local state, trigger side effects, and respond to changes in state, ensuring dynamic and responsive user interfaces.

Fetch API for API Requests:

API requests to the backend server are made using the Fetch API. This enables communication with the server to fetch data, submit forms. The integration with the backend server enables functionalities such as creating new posts and fetching existing ones.

Overall Application Functionality:

The combination of React.js, React Router DOM, state hooks, and Fetch API empowers the frontend to deliver a seamless and interactive user experience. Users can browse, create, and interact with content generated by the DALL-E AI model, facilitating the sharing and exploration of imaginative images within the application's community showcase.

8. BACK-END DESCRIPTION

Express.js and Node.js:

Express.js and Node.js provide the server-side environment and framework for handling HTTP requests and responses.They enable the creation of Restful APIs that allow clients to interact with the backend services.

In the context of AI image generation, Express.js routes can handle requests for generating new images, retrieving existing images, and managing user interactions.

MongoDB:

MongoDB serves as the database for storing generated images, user data, and other relevant information.

It offers flexibility in schema design, allowing the storage of image metadata, user preferences, and other related data in JSON-like documents.

MongoDB's scalability and performance make it suitable for handling large volumes of image data generated by AI models.

Backend Components:

Routes and Controllers: Express.js routes define endpoints for image generation, retrieval, and user interactions. Controllers contain the logic for handling these requests, including invoking AI models for image generation and interacting with the database to store or retrieve images.

Middleware: Middleware functions can be used for tasks such as authentication, request validation, and error handling. They ensure that incoming requests are properly processed and authorized before invoking the relevant controller functions.

Models: MongoDB models define the structure of documents stored in the database. In the context of AI image generation, models may represent images, users, sessions, or other entities relevant to the application. They provide methods for CRUD (Create, Read, Update, Delete) operations on the database, allowing efficient storage and retrieval of image data.

9. Results

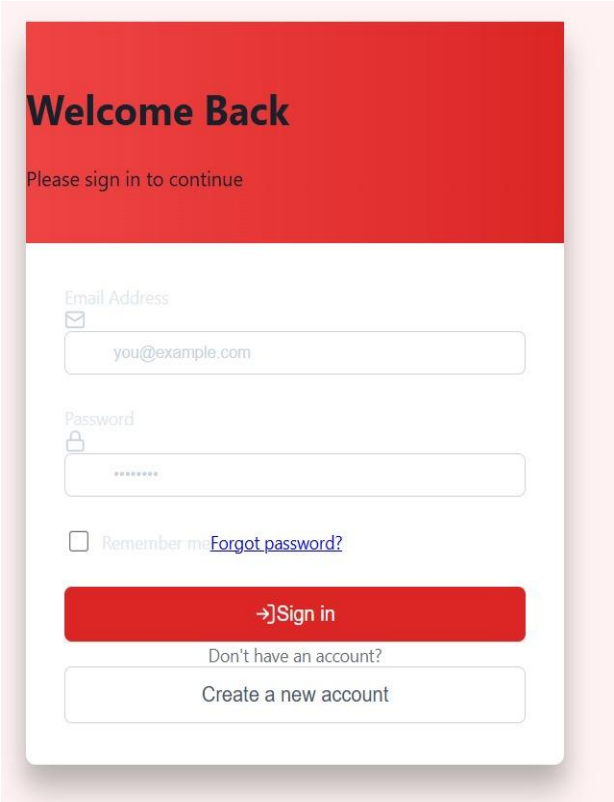


Figure-12:login page

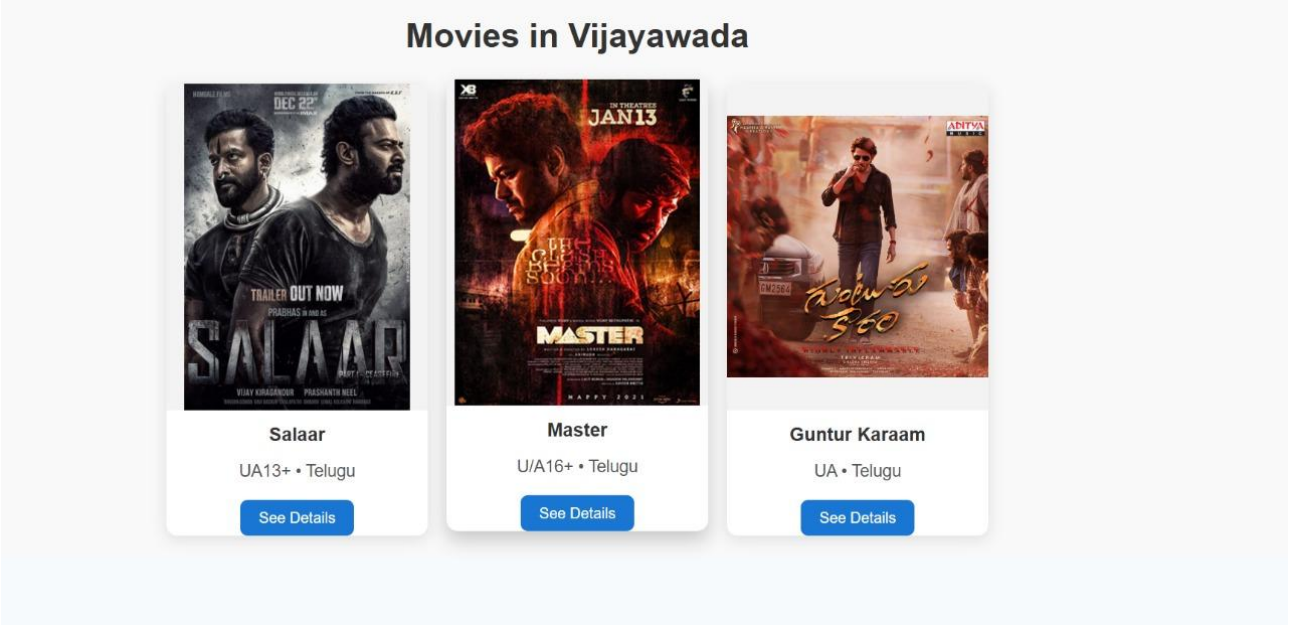
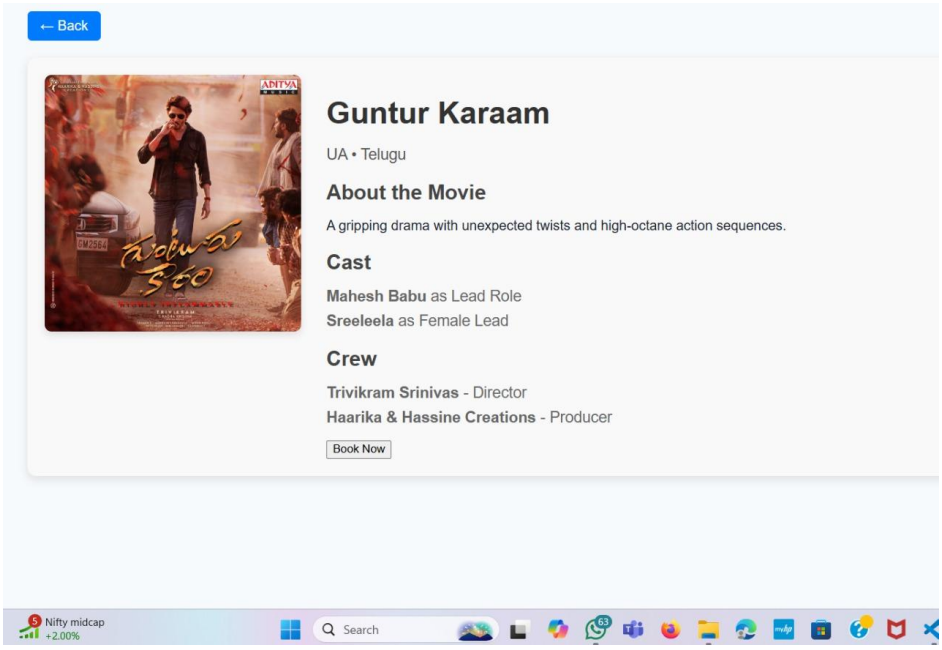


Figure-13: HomePage

Figure-14: Details of movie



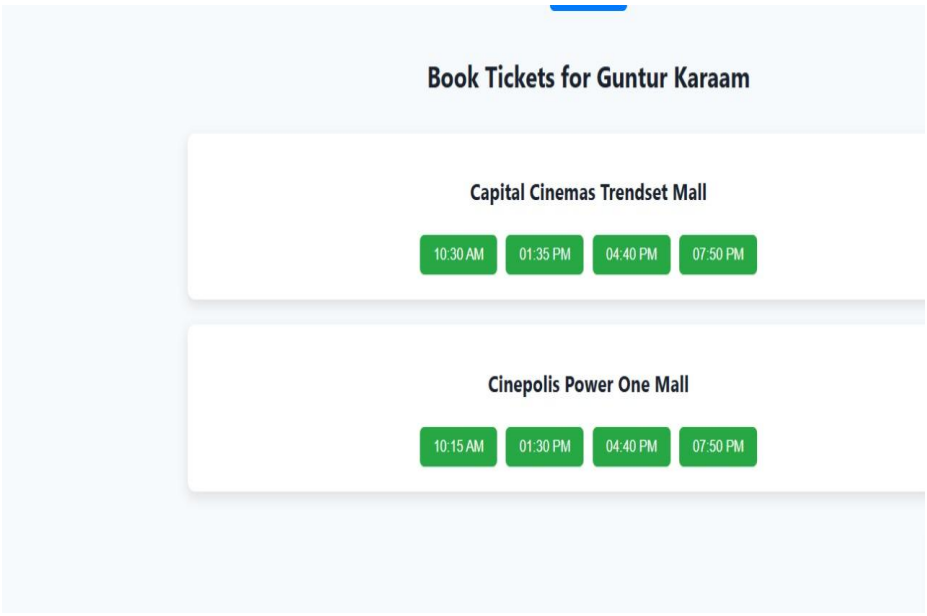


Figure-14: show times of movie

Figure-1.5:seats selection for movie

10. CONCLUSION

The integration of Movie booking systemgeneration within a technological framework comprising React.js for the front end, Node.js and Express.js for the back end, and MongoDB as the database presents a robust and scalable solution for modern applications. Through the seamless collaboration of these technologies, developers can harness the power of react.js algorithms to generate, manipulate, and optimize images dynamically, enhancing user experiences across various platforms.

The utilization of React.js on the front end ensures a responsive and interactive user interface, allowing for smooth integration of booking a movie within web applications. Meanwhile, the combination of Node.js and Express.js on the back end facilitates efficient handling of requests, processing of sever for booking, and seamless communication with the MongoDB database, enabling swift retrieval and storage of image data.

Moreover, MongoDB's flexible document-based structure accommodates the storage and retrieval of image-related information with ease, ensuring scalability and performance as the volume of data grows. This database solution complements the real-time capabilities of Node.js, facilitating rapid development and deployment of booking movies.