MOVIE BOOKING SYSTEM

A Project Report Submitted
to
MANIPAL ACADEMY OF HIGHER EDUCATION

For Partial Fulfillment of the Requirement for the
Award of the Degree
Of

Bachelor of Technology
in
Information & Communication Technology
by

Ankit Raj, Nilimp Vats, Tarikul Islam
210911256, 210911254, 210911064

Under the guidance of

Dr. Swathi BP (Lab Faculty 1)
Associate Professor
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

Dr. Girija Attigeri (Lab faculty 2)
Assistant Professor-Senior Scale
Department of I&CT
Manipal Institute of Technology
Manipal, Karnataka, India

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Unit of MAHE, Manipal

November 2023

ABSTRACT

The Movie Booking System (Cinematica) is an innovative movie booking system designed to streamline the process of booking movie tickets for users. Employing MySQL as the backend database, Flask as the server-side framework, and HTML, CSS, and JavaScript for the frontend, Cinematica provides a user-friendly interface for browsing movies, selecting showtimes, choosing seats, and completing bookings seamlessly.

Utilizing Flask as the backend facilitates efficient communication between the frontend and the MySQL database, ensuring robust data handling and retrieval. The system offers a dynamic and interactive experience, enabling users to directly access real-time information from the database, including available movie titles, show timings, theatre locations, seat availability, and booking details.

Cinematica's frontend, crafted with HTML, CSS, and JavaScript, offers an intuitive interface that allows users to navigate through movie listings, view showtime schedules, select preferred seats, and finalize bookings effortlessly. Leveraging the power of MySQL, the system ensures accurate and up-to-date information retrieval, presenting users with current and precise data for making informed booking decisions.

This comprehensive movie booking system seamlessly integrates backend functionalities with a user-centric frontend, delivering an engaging and responsive platform for users to explore movie options, reserve seats, and enjoy a hassle-free movie booking experience.

Cinematica, an advanced movie booking system, offers users a range of key features to enhance their movie booking experience. Users can conveniently select seats in their preferred city, choose from a list of preferred theaters, and pick their desired screen and timing for a seamless movie viewing experience. The platform allows users to select seats of their choice, ensuring a personalized and comfortable movie-watching experience. Cinematica supports various payment methods, including credit cards, debit cards, and gift cards, providing users with flexibility and convenience during the checkout process. Moreover, users have the freedom to cancel bookings at their convenience, offering flexibility and convenience in managing their movie plans.

In conclusion, Cinematica emerges as a comprehensive and user-centric movie booking solution, intertwining MySQL as the database backend, Flask as the server-side framework, and HTML, CSS, and JavaScript for an interactive frontend. Its core features empower users to effortlessly browse, select, and book movie tickets while providing the flexibility to choose preferred cities, theaters, screens, timings, and seats. The system's integration of multiple payment options and cancellation capabilities amplifies user convenience, ensuring a seamless and personalized movie booking journey. Cinematica's commitment to delivering real-time, accurate data directly from the database fosters trust and reliability, promising an engaging and efficient movie ticket reservation experience for all users.

INDEX

Chapter	Title
Chapter 1	INTRODUCTION
Chapter 2	LITERARY SURVEY/BACKGROUND
Chapter 3	OBJECTIVES/PROBLEM STATEMENT
Chapter 4	DATABASE DESIGN AND IMPLEMENTATION
Chapter 5	PROJECT ANALYSIS
Chapter 6	RESULTS
Chapter 7	CONCLUSION
Chapter 8	REFRENCES

LIST OF TABLES-

1) **Customer -** The Customer table holds important information about customers. It includes their name, username, email address, and password. This table is really important because it keeps track of all the details about customers and helps link their bookings. It helps make sure each customer gets a personalized experience when they book something.

2) **Movie -** The Movie table holds information regarding movies that are in the theatres. It keeps track of various details about each movie, such as the movie's name, how long it lasts (duration), what language it's in, its genre (like action, comedy, or drama), a brief description, its rating, and when it was released.This table is crucial because it stores essential information about the movies that people can watch at the theaters.

3) **City-** The City table holds information about different cities. It includes details such as the name of the city, the state it's located in, and its unique pincode.The City table is designed to facilitate user selection of a preferred city for watching a movie.

4) **Theatre -** The Theatre table lists different movie theaters in a chosen city for users to book tickets. It includes the theater names, how many screens each theater has, and the city they're located in.This table helps users see all the theaters available in the city they picked.

5) **Screen -** The Screen table stores information about screens within a specific theater. It includes details like the screen number and a reference to the theater it belongs to (identified by its theater ID).This table essentially keeps track of all the screens available within a particular theater.

6) **Seat -** The Seat table is where seat-related details are stored for booking purposes. It contains information such as the seat number, its status (whether it's available or booked), and the price associated with each seat.This table plays a crucial role in the booking process, allowing users to reserve specific seats. It keeps track of seat availability, indicating which seats are free for booking and which ones are already taken.

7) **Booking -** The Booking table acts as a repository for booked tickets and seat-related information. It contains essential details such as the customer ID, movie ID, city ID, theater ID, screen ID, showtime, seat ID, payment status, and date.This table is crucial for storing the details of reserved tickets, linking them to specific customers, movies, cities, theaters, screens, and seats. It maintains information about the booking status, including whether payment has been made or not, along with the date of the booking.

8) **Show_Time -** The Show_Time table contains information about showtimes associated with various screens. It comprises details such as the screen ID and the specific showtime for movies playing on those screens.

9) **Payment** - Payment Table conatins amount,payment method, transactionId.This table keeps a record for all the payments made by a user.

10) **Payment_Details** - This table keeps the record for the payments done by the user.

CHAPTER 1 – INTRODUCTION

Introducing Cinematica: Redefining Movie Booking Experiences -

In the fast-paced realm of entertainment, efficient and user-friendly movie booking systems have become integral to the cinema experience. Cinematica emerges as a revolutionary Movie Booking System designed to transform the way audiences reserve movie tickets. This innovative system seamlessly integrates HTML, CSS, and JavaScript for its engaging front-end, Flask as the robust back-end framework, and MySQL as the dependable database, promising a new era in movie ticket reservations.

Much like the streamlined operations of a modern restaurant facilitated by innovative technologies, Cinematica reimagines the process of movie ticket bookings. Combining advanced back-end functionalities with an intuitive front-end interface, it caters to the diverse needs of moviegoers, theater managers, and booking personnel. This system isn't just about simplifying bookings; it's a comprehensive solution aimed at enhancing user experiences, optimizing operations, and providing valuable data for strategic decision-making.

At its foundation, Cinematica relies on MySQL's reliability and efficiency as its database engine. This robust backend infrastructure forms the backbone of the system, handling movie listings, seat allocations, booking details, and user preferences, ensuring a well-organized cinema operation management system.

The front-end interface of Cinematica, powered by HTML, CSS, and JavaScript, boasts simplicity and sophistication. It's designed to provide an engaging and user-friendly experience, ensuring clear information display, ease of navigation, and interactive elements for hassle-free movie ticket reservations.

Facilitating real-time capabilities and seamless communication between the front-end and back-end, Flask drives Cinematica's backend operations. This integration enables swift updates, instant ticket bookings, and dynamic interactions, solidifying Cinematica as a cutting-edge solution in the domain of movie ticketing systems.

Cinematica represents more than just a software application; it's a transformative tool tailored to meet the evolving demands of the entertainment industry. From managing movie listings to streamlining booking processes and delivering valuable insights, Cinematica empowers theaters to provide a seamless, engaging, and efficient movie-going experience.

In essence, Cinematica symbolizes the convergence of HTML, CSS, JavaScript, Flask, and MySQL technologies. Its commitment to enhancing the movie ticketing experience, operational efficiency, and user satisfaction makes it an essential asset for cinemas aiming to thrive in today's competitive market.

1.1 OUTLINE-

The movie booking system offers a comprehensive and user-friendly platform for cinephiles and casual moviegoers alike. Users are welcomed with a seamless registration process , ensuring convenient access to the system. The interface boasts an extensive movie library categorized by genre, language, and release date, allowing patrons to browse through detailed movie descriptions and cast information.The movie booking system redefines convenience by enabling users to select showtimes and theaters effortlessly. Through an intuitive interface, users can navigate a comprehensive list of available showtimes and theaters within their chosen city, selecting their preferred options seamlessly. The absence of a cart system streamlines the booking process, allowing users to proceed directly to confirm their bookings without multiple steps. Upon selecting the desired city, theater, and showtime, users can choose their preferred seats. The straightforward payment process completes the transaction, generating a unique QR code that serves as a digital ticket. To manage their bookings, users can easily access and review their confirmed tickets within the 'My Booking' section. This system ensures a hassle-free experience, simplifying the movie-going process by integrating city, theater, showtime, and seat selection into a streamlined booking journey. The digital ticketing system, empowered by QR codes, enhances convenience and efficiency for users while enjoying their cinematic experience.

1.2 CONTEXT-

In an era where entertainment choices abound and time is a precious commodity, the Movie Booking System emerges as a beacon of simplicity and efficiency. With a vision to transform the traditional movie ticketing experience, this system stands as a testament to streamlined convenience. In a bustling cityscape, where individuals seek solace or joy within the confines of a theater, this platform becomes a trusted ally. Imagine a world where the arduous process of selecting a movie, finding a suitable showtime, and securing desired seats is no longer a labyrinth of endless clicks and screens. Instead, it's an intuitive journey, where users effortlessly traverse through a comprehensive list of showtimes and theaters at their preferred city's fingertips. Here, the absence of a cart system doesn't mean complexity but rather liberation from unnecessary steps. Users seamlessly select their city, theater, and showtime, their anticipation growing as they pick their favorite seats. A swift payment process culminates in the generation of a digital QR code, unlocking the gateway to their cinematic adventure. This system doesn't just offer a ticket; it crafts an experience, where every interaction is purposeful, and every moment, cherished. With a dedicated 'My Booking' section, users can effortlessly track and manage their confirmed tickets, ensuring peace of mind and fostering excitement for the upcoming movie night. This innovation doesn't just cater to moviegoers; it speaks to the modern pace of life—a commitment to simplicity, efficiency, and an unwavering dedication to enhancing every facet of the entertainment experience.

CHAPTER 2- LITERARY SURVEY/BACKGROUND

The movie industry is a multi-billion dollar industry that is constantly evolving. In recent years, there has been a significant shift towards online movie booking systems. These systems offer a number of advantages to both customers and movie theaters. For customers, online booking systems provide convenience, flexibility, and the ability to compare prices. For movie theaters, online booking systems can reduce costs, increase efficiency, and improve customer satisfaction.

An online movie booking system is a software application that allows customers to book movie tickets online. The system typically includes a database of movies, showtimes, and seats. Customers can browse the database to find the movie they want to see, and then select the showtime and seats they want. The system will then process the payment and send the customer an e-ticket.

There are a number of benefits to using an online movie booking system. For customers, the benefits include:
Convenience: Customers can book tickets from anywhere, at any time.
Flexibility: Customers can compare prices and find the best deals.
Information: Customers can read reviews and see trailers before they book tickets.
Avoidance of queues: Customers can avoid waiting in line at the box office.
For movie theaters, the benefits include:
Reduced costs: Online booking systems can reduce labor costs.
Increased efficiency: Online booking systems can process transactions more quickly.
Improved customer satisfaction: Online booking systems can provide a more convenient and enjoyable experience for customers.

A number of studies have been conducted on the impact of online movie booking systems. These studies have found that online booking systems have a number of positive effects on both customers and movie theaters. For example, a study by Chen and Wang (2010) found that online booking systems can increase customer satisfaction and loyalty. Another study, by Lee and Park (2012), found that online booking systems can improve the efficiency of movie theater operations.

Online movie booking systems are a valuable tool for both customers and movie theaters. They offer a number of benefits, including convenience, flexibility, and efficiency. Online booking systems are a growing trend in the movie industry, and they are likely to continue to be popular in the years to come.

CHAPTER 3- OBJECTIVES/PROBLEM STATEMENT

3.1 Problem Statement-

In the movie industry, existing ticket booking systems face challenges in managing sales, optimizing seat availability, and meeting evolving user expectations. Manual processes contribute to inefficiencies, errors, and a subpar customer experience. This results in revenue loss and an inability to adapt to modern moviegoers' needs.
To tackle these issues, there's a critical need for a digital solution. The project aims to create an Movie Booking System (OMTBS) using advanced database design and a user-friendly interface. This solution will not only streamline ticket booking processes but also enhance customer satisfaction, boost ticket sales, and reduce operational costs in the movie booking landscape.

Objectives-

1. **Customer-Centric Functionality:** Enable users to seamlessly explore, book, and purchase movie tickets.

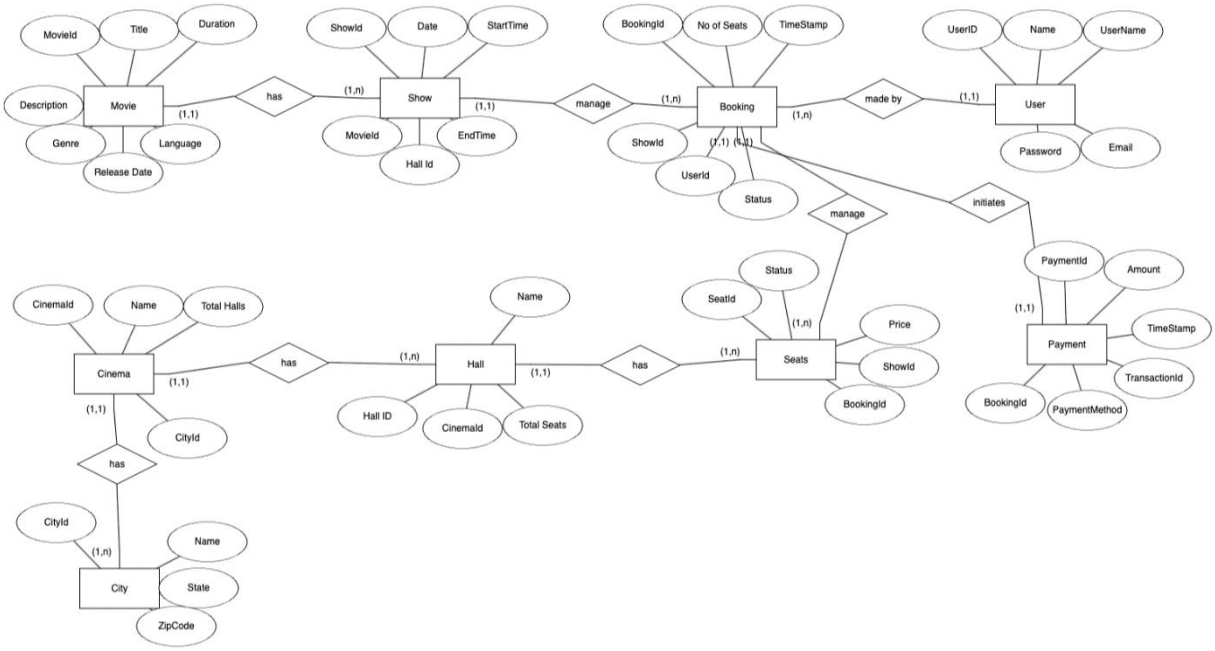
2. **Registration and Account Management:** Implement user registration and account management features tailored for moviegoers.

3. **Booking Management**: Develop a robust system for placing, tracking, and managing movie ticket bookings.

4. **Seat and Show Management**: Design and implement features for administrators to manage seat availability and movie show schedules.

CHAPTER 4- DATA DESIGN

4.1 ER DIAGRAM



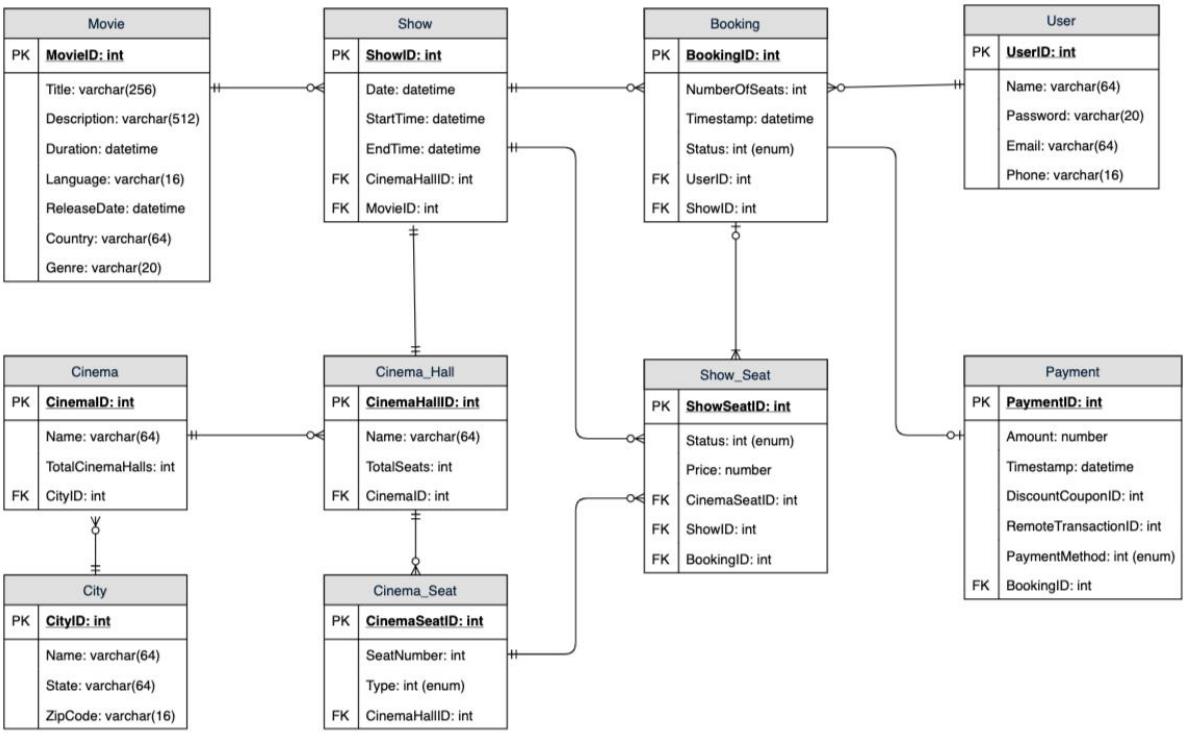
4.2 Reduction

The reduction phase in the database design process for the 'Movie Booking System' (MBS) involves simplifying the intricate Entity-Relationship (ER) diagram into a streamlined and focused representation. This process aims to underscore vital components and relationships, ultimately improving the clarity and comprehension of the database structure. By condensing the ER diagram, we strive to create a more succinct and manageable view, emphasizing the core entities and their interconnections within the system. The ensuing sections present a refined depiction of the primary database tables, offering a clearer insight into the structure of the MBS.

Movie (movieid, title, description, duration, Language, release date, genre)
Show (showid, date, starttime, endtime, theatreid, movieid)
Booking (bookingid, noofseats, status, userid, showid)
User (userid, name, password, email, username)
Cinema (cinemaid, name, cityid, noofhalls)
Theatre (theatreid, name, totalseats, cinemaid)
Seat (seatid, status, theatreid)
Payment (paymentid, amount, transactionid, paymentmethod, bookingid)
City (cityid, name, state)

Now, each table represents a single entity, and relationships between tables are defined by foreign keys. This normalization ensures that data is stored efficiently, with minimal redundancy and potential for update anomalies.

4.3 SCHEMA Diagram



4.4 Normalization

First Normal Form (1NF)

In 1NF, each attribute of a table must contain only atomic (indivisible) values, and there should be no repeating groups or arrays.

Movie (movieid, title, description, duration, Language, release date, genre):
Show (showid, date, starttime, end time, cinemahallid, movieid):
Booking (bookingid, noofseats, status, userid, showid):
User (userid, name, password, email, username):
Cinema (cinemaid, name, cityid, noofhalls):
Cinemahall (cinemahallid, name, total seats, cinemaid):
Seat (seatid, seatrow, type, cinemahallid):
Payment (paymentid, amount, transactionid, payment method, bookingid):
City (cityid, name, state):

All attributes contain atomic values, and there are no repeating groups.

Second Normal Form (2NF)

In 2NF, a table must be in 1NF, and all non-prime attributes should be fully functionally dependent on the entire primary key.

1. Movie (movieid, title, description, duration, Language, release date, genre):
   All attributes are fully functionally dependent on the primary key (movieid). No partial dependencies exist.

2. Show (showid, date, starttime, end time, cinemahallid, movieid):
   The composite key is (showid, cinemahallid). All non-key attributes (starttime, endtime, movieid) are fully functionally dependent on the entire composite key.

3. Booking (bookingid, noof seats, status, userid, showid):
   All attributes are fully functionally dependent on the primary key (bookingid).

4. User (userid, name, password, email, username):
   All attributes are fully functionally dependent on the primary key (userid).

5. Cinema (cinemaid, name, cityid, noofhalls):
   All attributes are fully functionally dependent on the primary key (cinemaid).

6. Cinemahall (cinemahallid, name, total seats, cinemaid):
   All attributes are fully functionally dependent on the primary key (cinemahallid).

7. Seat (seatid, seatrow, type, cinemahallid):
   All attributes are fully functionally dependent on the primary key (seatid).

8. Payment (paymentid, amount, transactionid, payment method, bookingid):
   All attributes are fully functionally dependent on the primary key (paymentid).

9. City (cityid, name, state):
   All attributes are fully functionally dependent on the primary key (cityid).

Third Normal Form (3NF):

In 3NF, a table must be in 2NF, and it should not have transitive dependencies.

1. Movie (movieid, title, description, duration, Language, release date, genre):
   No transitive dependencies exist; all attributes are directly dependent on the primary key.

2. Show (showid, date, starttime, end time, cinemahallid, movieid):
   The attributes starttime and endtime depend only on the composite key (showid, cinemahallid). To remove the transitive dependency, we can create a new table for movie-related information.

3. MovieShow (showid, cinemahallid, movieid)

4. Booking (bookingid, noof seats, status, userid, showid):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

5. User (userid, name, password, email, username):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

6. Cinema (cinemaid, name, cityid, noofhalls):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

7. Cinemahall (cinemahallid, name, total seats, cinemaid):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

8. Seat (seatid, seatrow, type, cinemahallid):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

9. Payment (paymentid, amount, transactionid, payment method, bookingid):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

10. City (cityid, name, state):
No transitive dependencies exist; all attributes are directly dependent on the primary key.

BCNF (Boyce-Codd Normal Form) :

Movie (movieid, title, description, duration, Language, release date, genre): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

Show (showid, date, starttime, end time, cinemahallid, movieid): The table is not in BCNF because there is a non-trivial functional dependency between {showid, cinemahallid} and movieid. To bring it into BCNF, we can decompose it into two tables:
Show (showid, date, starttime, endtime, cinemahallid)
MovieShow (showid, cinemahallid, movieid)

Booking (bookingid, noof seats, status, userid, showid): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

User (userid, name, password, email, username): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

Cinema (cinemaid, name, cityid, noof halls):This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

Cinemahall (cinemahallid, name, total seats, cinemaid): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

Seat (seatid, seatrow, type, cinemahallid): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

Payment (paymentid, amount, transactionid, payment method, bookingid): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

City (cityid, name, state): This table is in BCNF. All non-prime attributes are fully functionally dependent on the primary key.

CHAPTER 5- Methodology

To develop theMovie Booking System (MBS), a systematic and comprehensive methodology was followed, encompassing stages such as data gathering, schema design, and user interface (UI) creation. This section provides an overview of the step-by-step process undertaken to achieve the desired outcomes.

1. Data Gathering:

Initial data gathering involved understanding the requirements of the movie booking system.

2. Schema Design:

Based on the gathered data, a detailed database schema was formulated to represent the various entities and their relationships within the movie booking system. Entities, such as movies, users, bookings, cinemas, and payments, were identified and organized into separate tables, ensuring that attributes and relationships were accurately defined.

The schema design included primary keys, foreign keys, and necessary constraints to maintain data integrity. For instance, the Movie table contained fields like MovieID, Title, and Genre, while the Bookings table included attributes such as BookingID, UserID, and ShowID. The schema aimed to capture all the essential information needed for effective movie ticket booking management.

3. User Interface (UI) Creation:

Simultaneously with schema design, emphasis was placed on developing an intuitive and user-friendly UI for the MBS. The UI was crafted using a combination of HTML, CSS, JavaScript, and the Bootstrap framework to facilitate easy navigation and efficient data entry.

The UI design featured a visually appealing layout with well-organized sections for different functionalities. Elements were created to support seamless data entry, retrieval, and manipulation. CSS styling enhanced visual presentation, while

JavaScript and Bootstrap were employed to add functionality and responsiveness to the UI.

4. Normalization:

In order to ensure data integrity, eliminate redundancy, and optimize database performance, normalization principles were applied to the tables in the movie booking system. The normalization process involved analyzing data requirements and applying normalization principles, specifically up to the third normal form (3NF). Tables were broken down into smaller, focused entities, and relationships were established to reduce redundancy.

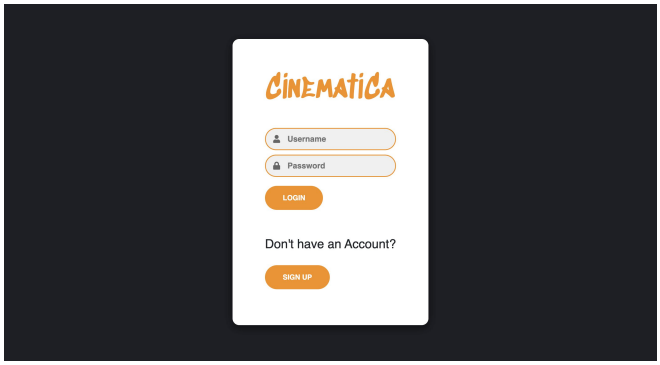
5. Integration:

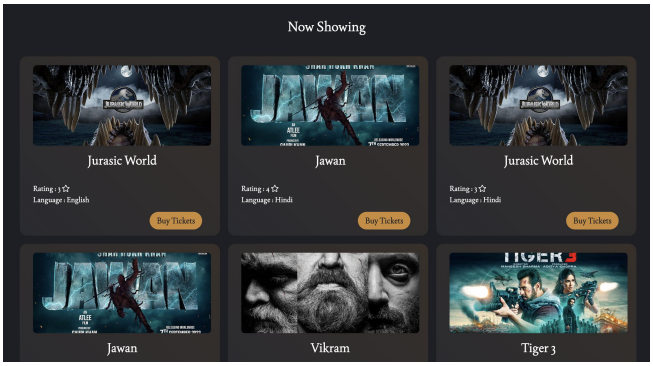
The methodology culminated in the seamless integration of database design and UI creation. The insights gathered from stakeholders informed the creation of a well-structured database schema, enabling efficient data management. Concurrently, the UI design process aimed at delivering a user-friendly interface that met the needs of moviegoers.

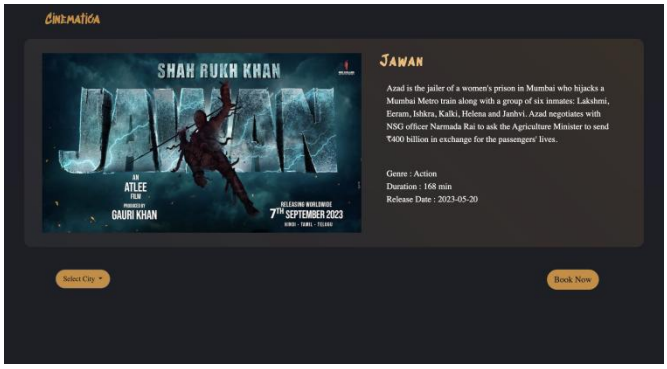
6. Outcome:

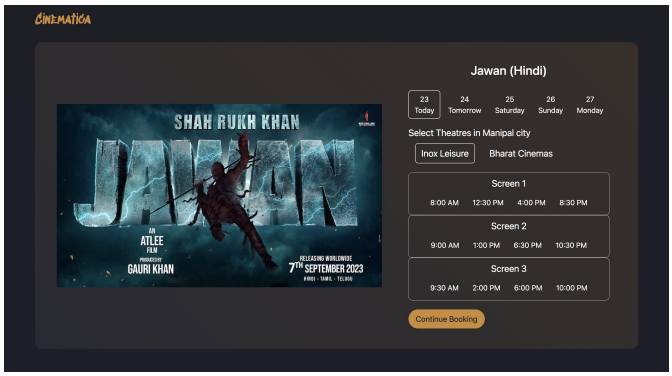
The systematic application of this methodology resulted in the development of a robust Online Movie Ticket Booking System. The integrated database and user interface lay the foundation for an effective system, streamlining movie ticket booking, show tracking, and enhancing the overall moviegoing experience for users.

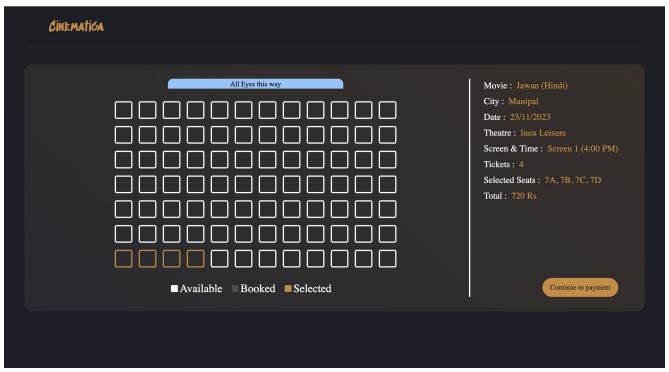
CHAPTER 6- RESULTS

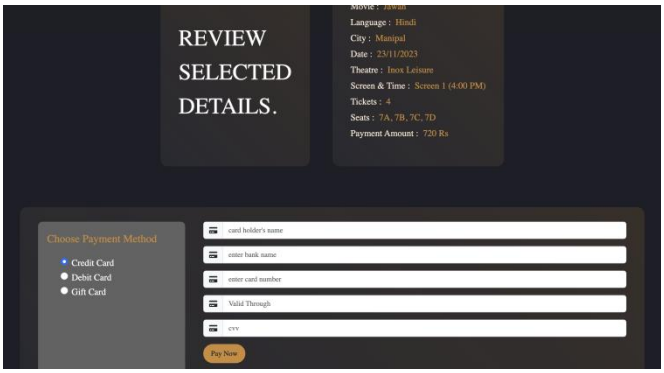


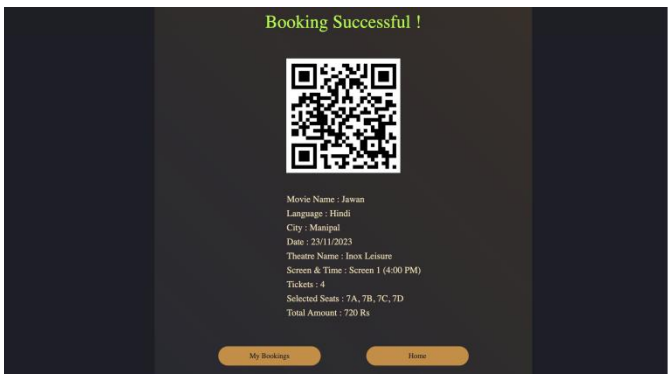


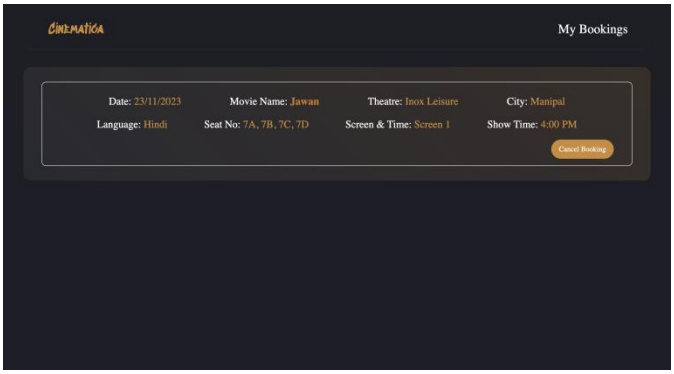












CHAPTER 7 – CONCLUSIONS

In summary, the development of the Movie Booking System (MBS) has followed a meticulous methodology, resulting in a well-structured database schema and a user-friendly interface. By applying normalization principles, the system ensures data integrity and optimal performance.

The integration of a responsive UI, designed with HTML, CSS, JavaScript, and Bootstrap, enhances the overall moviegoing experience. The MBS not only holds academic value but also provides a practical solution that caters to the changing needs of movie enthusiasts.

In conclusion, the project successfully streamlines movie ticket booking, show tracking, and customer relations, presenting a promising and efficient platform for users in the digital age.

7.1 FUTURE SCOPE

Future improvements for MBS could involve integrating predictive analytics using machine learning algorithms to optimize seat availability. Additionally, exploring e-commerce integration would offer an omnichannel movie booking experience, incorporating online merchandise sales and exclusive offers. The system's scope could be expanded to include supply chain management features, enhancing coordination with cinemas and distributors for more efficient order fulfillment. Continuous enhancements to the user interface would focus on refining navigation and introducing innovative features, ensuring a user-friendly platform that aligns with evolving industry trends. These developments aim to increase overall efficiency and responsiveness in managing movie booking operations.

CHAPTER 8- REFERENCES

8.1 Websites

- https://www.geeksforgeeks.org/
- https://www.w3schools.com/
- https://www.tutorialspoint.com/
- https://www.programiz.com/