Database Management Lab

Online Movie Booking System

VU21CSEN0101244 – T.Krishna
VU21CSEN0100089 – M.Tanusha
VU21CSEN0101845 – S.Venkatapathi Varma

Prof. K. Thammi Reddy

Department of Computer Science and Engineering
GITAM School of Technology, GITAM (Deemed to be University)
Visakhapatnam

November - 2023

Contents

1. Introduction about the Use case
2. Requirement analysis considered for the use case
    a.) Software requirements
    b.) Hardware requirements
3. ER- model
4. Transforming ER to Relational model
5. Desing issues: Normalization
6. Database connectivity using a front end
7. Conclusion
8. References

1.Introduction about the Use case :

Project Overview :

Welcome to the case study documentation of our project, which revolves around the creation of an Online Movie Ticket Booking System. This system represents a crucial endeavor in the modernization of the movie ticket booking process, aiming to provide moviegoers with a convenient and efficient platform to access movie listings, view showtimes, and book tickets online. The project employs a synergistic blend of various technologies, including HTML, CSS, JavaScript, and the robust Oracle Database Management System (DBMS).

Project Objectives

Our project is driven by several core objectives:

Streamlined Booking Process: The primary goal is to revolutionize the booking process by making it more accessible and user-friendly. We aim to eliminate the complexities and frustrations often associated with ticket booking, ensuring that users have a hassle-free and enjoyable experience.

User-Friendly Interface: The heart of the system is an intuitive and visually appealing web interface. Users should effortlessly navigate through movie listings, explore showtimes, and select seating options. We pay meticulous attention to design, ensuring that our interface is both aesthetically pleasing and easy to use.

Data Management: The project places strong emphasis on efficient data management. We employ the Oracle DBMS to securely store and manage an array of crucial data, including movie details, theater information, showtimes, and customer profiles. This guarantees data integrity, reliability, and scalability.

Integration of Technologies: The project's foundation rests on the integration of front-end technologies (HTML, CSS, JavaScript) with the power of an Oracle DBMS on the back end. This integration is the key to creating a seamless, dependable, and feature-rich application.

2. Required Analysis :

Certainly, for an online movie ticket booking system, you would need to consider specific software and hardware requirements during the requirement analysis phase. Here's what you might consider :

a.) Software Requirements:

1).Operating System: Determine which operating systems the system should be compatible with. For example, it might need to support Windows, macOS, and various mobile operating systems like Android and iOS.

2). Web Development Framework: Identify the web development framework or platform you'll use for building the online booking system, such as Ruby on Rails, Django, or a JavaScript framework like React.

3).Database Management System: Specify the database management system for storing and retrieving data related to movies, showtimes, and reservations. Common choices include MySQL, PostgreSQL, or NoSQL databases like MongoDB.

4).Payment Gateway Integration: If the system involves online payments for ticket bookings, you'll need to integrate a payment gateway such as PayPal, Stripe, or a bank's payment processing API.

5).Responsive Web Design: Ensure that the user interface is designed to be responsive, adapting to different screen sizes and devices.

6). APIs and Integrations: If you plan to offer features like movie showtime data from theaters or integration with external services (e.g., movie reviews), specify the APIs and integrations required.

b). Hardware requirements:

1).Server Infrastructure: Determine the server specifications needed to host the website and manage the database. Consider factors like CPU, RAM, and storage capacity.

2). Redundancy and Failover: Implement redundancy measures, such as backup servers or cloud infrastructure, to ensure high availability and fault tolerance.

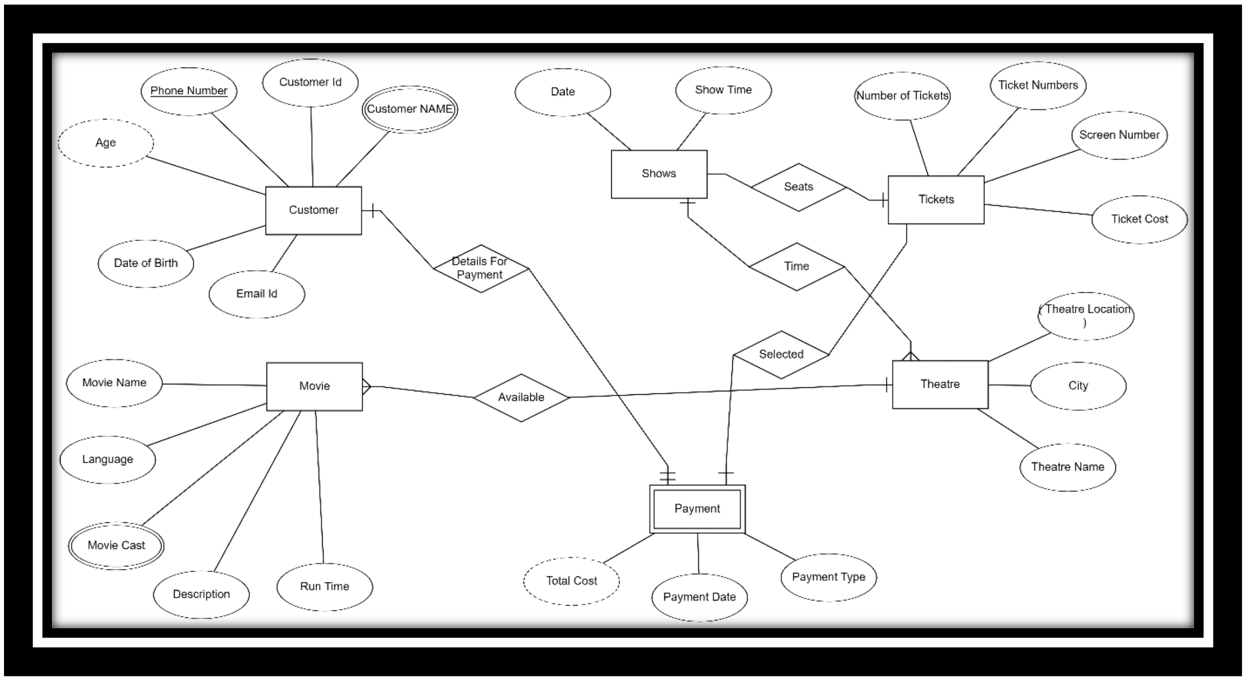
3). Load Balancing: Set up load balancing mechanisms to evenly distribute incoming traffic across multiple servers for improved performance.

4).Network Infrastructure: Ensure a reliable and high-speed internet connection for both the servers and users, as network speed and stability are crucial for online services.

5). Client Devices: Consider the hardware and software capabilities of the devices your users will use to access the system. Ensure the website is compatible with various web browsers and devices, including desktops, laptops, smartphones, and tablets.

6).Ticketing Hardware: If the system integrates with physical ticketing at movie theaters (e.g., barcode scanners), define the hardware and software requirements for these components.

3.ER Model



4.Transforming ER to Relational model

Using the E-R model, we can create logical tables for the database:

CUSTOMER (CustomerID [PK], CustomerName, Email, Phone, Address,DOB)

```
CREATE TABLE Customer(
 CustomerID INT PRIMARY KEY,
 CustomerName VARCHAR(255),
 PhoneNo VARCHAR(255),
 DateOfBirth DATE,
 EmailID VARCHAR(255) );
desc customer;
insert into customer values(100,'john',9911991199,'20-09-1998','john@gmail.com');
insert into customer values(102,'Smith',9911223344,'21-05-2000','alice@gmail.com');
insert into customer values(103, 'Bob', 6677889911, '20-02-2001', 'bob@mail.com');
insert into customer values(104, 'Jack', 8099112233, '25-01-1995', 'jack@gmail.com');
insert into customer values(105, 'Shaq', 9919223344, '19-12-2002', 'shaq@gmail.com');
insert into customer values(106, 'Bard', 6687889911, '29-10-2010', 'bard@mail.com');
```

Movie (MovieID, MovieName, Language, MovieCast , Description, Runtime)

```
CREATE TABLE Movie ( MovieID INT PRIMARY KEY, MovieName VARCHAR(255), Languag VARCHAR(255), MovieCast
VARCHAR(100), Description VARCHAR(300) , Runtime INT );
insert into Movie values (1000, 'WOLF', 'English', 'DON,BRAD', 'ACTION', 120);
 insert into Movie values (1002, 'NARCOS', 'Spanish', 'PIT,SELENA', 'ADVENTUTURE',135);
insert into Movie values (1003, 'PATRO', 'French', 'MAX,KIM', 'ROMCOM', 110);
 insert into Movie values (1004, 'HITLER', 'German', 'HOLA,SAKA', 'SITCOM', 125);
insert into Movie values(1005, 'SUII', 'Italian', 'PELE,KANTE', 'ROMANCE', 140);
insert into Movie values (1006, 'ONE', 'Japanese', 'JOKER,NISA', 'HORROR', 115);
```

Theater (TheaterID [PK], TheaterName, Location, City)

```
CREATE TABLE Theater( TheaterID INT PRIMARY KEY, TheaterLocation VARCHAR(255), City VARCHAR(255),
TheaterName VARCHAR(255) );
Insert into Theater values(11, 'UPPAL', 'HYD', 'INOX');
Insert into Theater values(12, 'BANDRA', 'CST', 'CINEPOLIS');
Insert into Theater values(13, 'ENDADA', 'VIZAG', 'NEETRA');
Insert into Theater values(14, 'KAROLBAGH', 'DELHI', 'SUDHARSHAN');
Insert into Theater values(15, 'WORLI', 'MUMBAI', 'IMAX');
Insert into Theater values(16, 'NADA', 'YANAM', 'PVR'); Select * from Theater;
```

Showtime (ShowtimeID [PK, TheaterID [FK], ShowTime, ShowDate,)

```
CREATE TABLE Show ( ShowID INT PRIMARY KEY, ShowDate DATE, ShowTime INT, TheaterID INT, FOREIGN
KEY (TheaterID) REFERENCES Theater(TheaterID) );
insert into Show values (1, '20-oct-2023', 14, 11);
 insert into Show values (2, '11-jan-2023', 16, 12);
insert into Show values (3, '01-oct-2023', 15, 11);
 insert into Show values (4, '08-dec-2023', 18, 13);
 insert into Show values (5, '28-sep-2023', 17, 15);
 insert into Show values (6, '30-feb-2023', 20, 16);
```

Ticket (TicketID [PK], CustomerID [FK], ShowtimeID [FK], TicketNumber, number of tickets,Screen number,Total Cost)

```
CREATE TABLE Ticket( TicketID INT PRIMARY KEY, NumberOfTickets INT, TicketNumbers
VARCHAR(255), ScreenNumber INT, TotalCost DECIMAL(10, 2), ShowID INT, CustomerID INT,
FOREIGN KEY (ShowID) REFERENCES Show(ShowID), FOREIGN KEY (CustomerID) REFERENCES
Customer7(CustomerID) );
insert into Ticket values (211, 2, 'A1, A2', 1, 200.00, 1, 100);
insert into Ticket values (112, 3, 'B3, B4, B5', 2, 300.00, 2, 102);
insert into Ticket values (223, 1, 'C1', 3, 100.00, 3, 103);
insert into Ticket values (334, 4, 'D1, D2, D3, D4', 1, 400.00, 4, 104);
insert into Ticket values (355, 2, 'E5, E6', 2, 220.00, 5, 105);
insert into Ticket values (556, 3, 'F3, F4, F5', 3, 330.00, 5, 106);
```

Payment (PaymentID [PK], Total Cost ,CustomerID [FK],, PaymentDate, PaymentType, TicketID [FK])

```
CREATE TABLE Payment ( PaymentID INT PRIMARY KEY, TotalCost DECIMAL(10, 2), PaymentDate DATE,
PaymentType VARCHAR(255), TicketID INT, FOREIGN KEY (TicketID) REFERENCES Ticket(TicketID) );
 insert into Payment values (21, 200.00, '20-oct-2023', 'Credit Card', 211);
insert into Payment values (22, 300.00, '11-jan-2023', 'PhonePay', 112);
insert into Payment values (32, 100.00, '01-oct-2023', 'Debit Card',223);
insert into Payment values (42, 400.00, '08-dec-2023', 'Paytm',334); i
nsert into Payment values (52, 220.00, '28-sep-2023', 'GPay',223);
insert into Payment values (62, 330.00, '22-dec-2023', 'Debit Card',556);
```

SQL Queries:

SET COMPARISION :

/*1.Print Payment Id, Payment Date , Payment Time where Pid is in between 22 and 52.*/

SELECT PaymentID, PaymentDate, PaymentType

FROM Payment

```
WHERE PaymentID >= 22 AND PaymentID <= 52;
```

/*2.Print MovieName , Movie cast and description where runtime is less than 130 mins..*/

```
SELECT MovieName, MovieCast, Description
FROM Movie
WHERE Runtime < 130;
```

NESTED QUERIES:

/*1.Print Customer Name where Ticket Id is 355.*/

```
SELECT CustomerName
FROM Customer c
WHERE EXISTS (
    SELECT *
    FROM Ticket T
    WHERE TicketID = 355 AND T.CustomerID = c.CustomerID
);
```

AGGREGATE OPERATORS:

/*1.Print the average cost of tickets sold..*/

```
SELECT AVG(TotalCost) FROM Ticket;
```

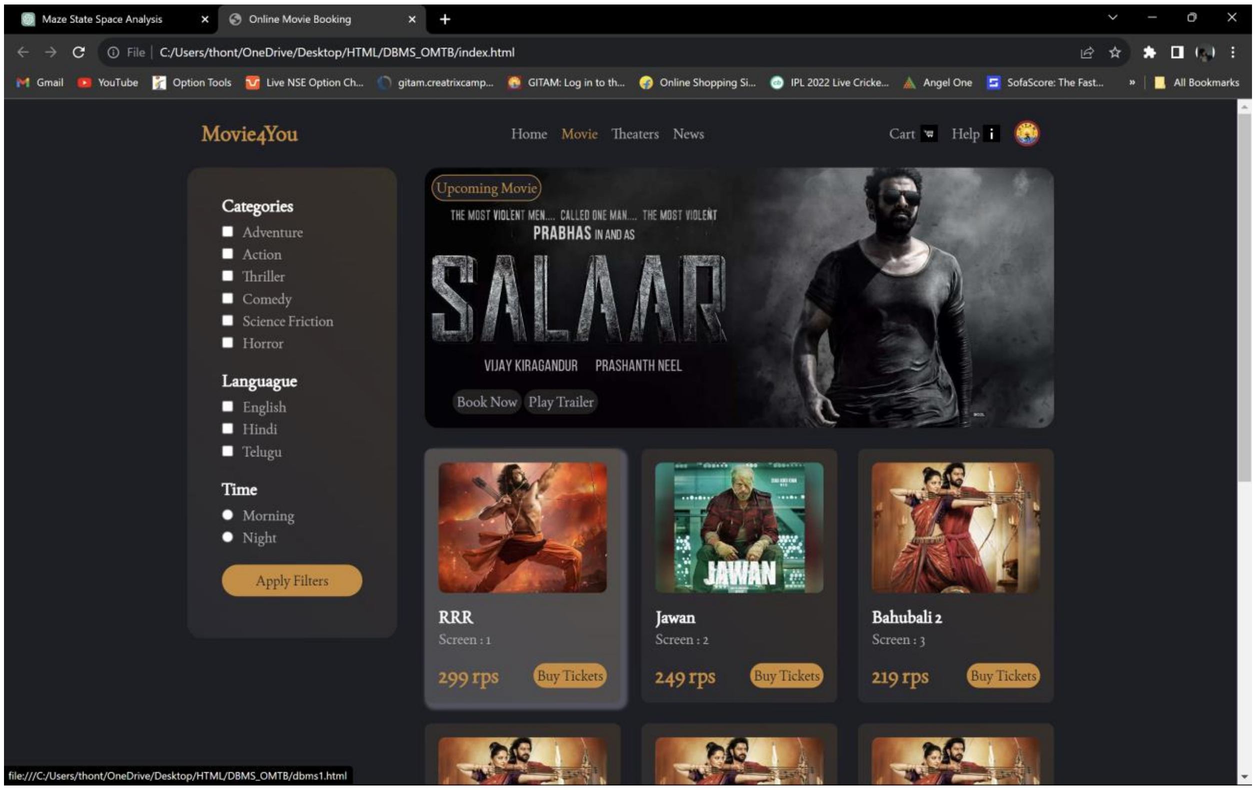
/*2.Print the movie name and runtime where the runtime is the highest.*/

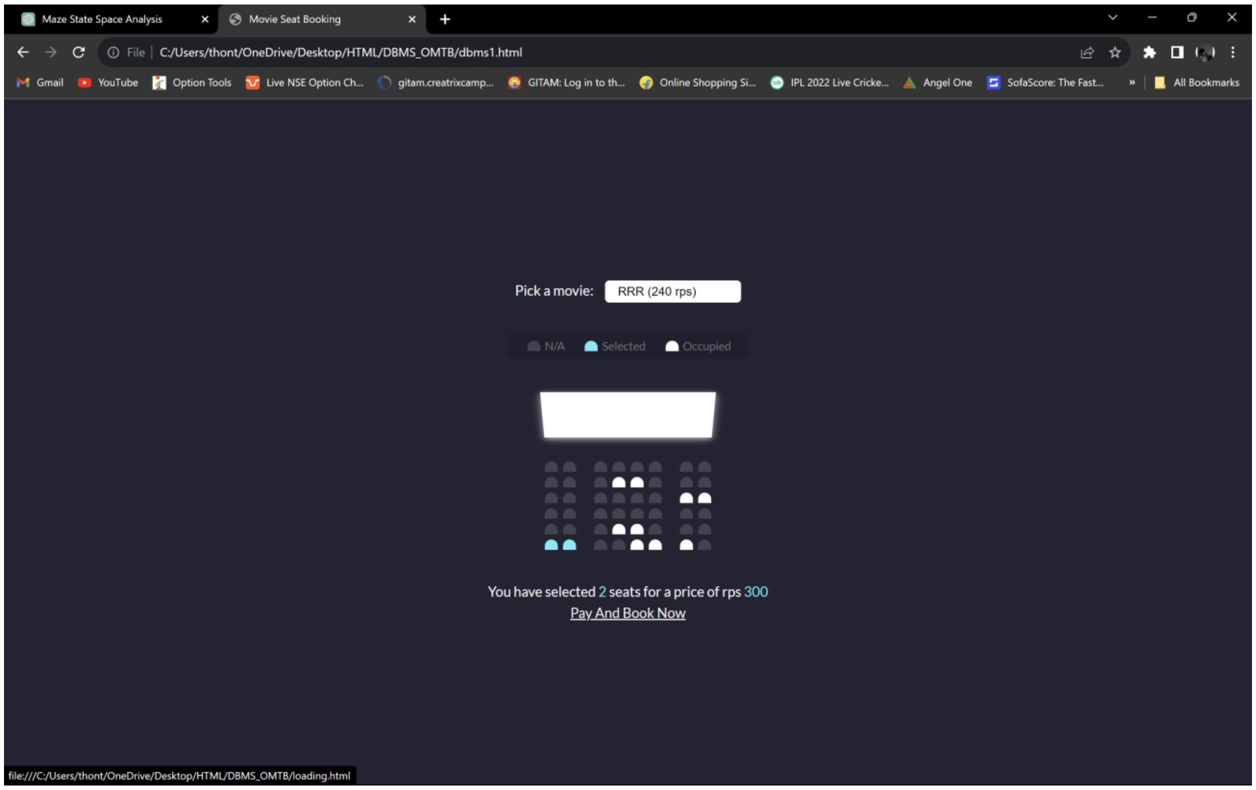
```
SELECT MovieName, Runtime
FROM Movie
WHERE Runtime = (SELECT MAX(Runtime) FROM Movie);
```

/*3.Print the number of customers in the database.*/

```
SELECT COUNT(CustomerID) FROM Customer;
```

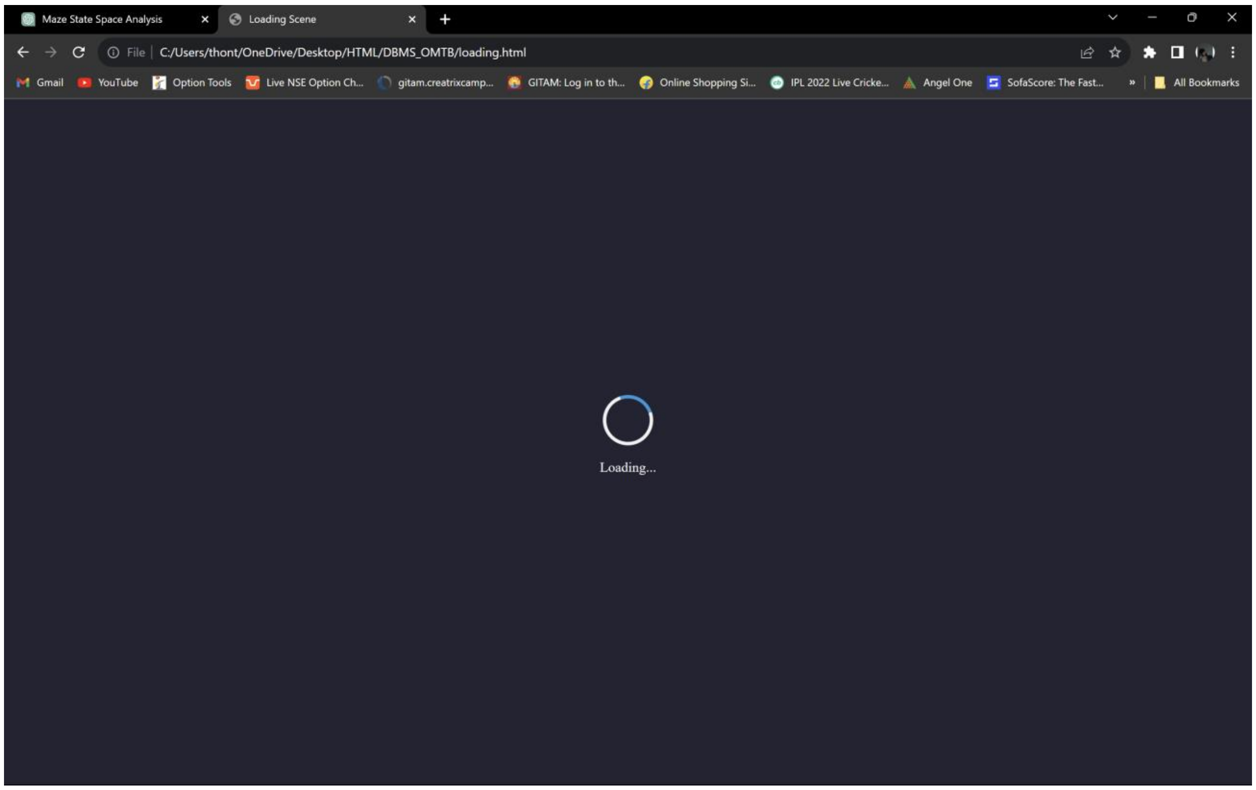
Front End :


Maze State Space Analysis
Online Movie Booking
File | C:/Users/thont/OneDrive/Desktop/HTML/DBMS_OMTB/index.html
Gmail
YouTube
Option Tools
Live NSE Option Ch...
gitam.creatrixcamp...
GITAM: Log in to th...
Online Shopping Si...
IPL 2022 Live Cricke...
Angel One
SofaScore: The Fast...
All Bookmarks
Movie4You
Home
Movie
Theaters
News
Cart
Help
Categories
Adventure
Action
Thriller
Comedy
Science Friction
Horror
Language
English
Hindi
Telugu
Time
Morning
Night
Apply Filters
Upcoming Movie
THE MOST VIOLENT MEN.... CALLED ONE MAN.... THE MOST VIOLENT
PRABHAS IN AND AS
SALAAR
VIJAY KIRAGANDUR PRASHANTH NEEL
Book Now
Play Trailer
RRR
Screen : 1
299 rps
Buy Tickets
Jawan
Screen : 2
249 rps
Buy Tickets
Bahubali 2
Screen : 3
219 rps
Buy Tickets
file:///C:/Users/thont/OneDrive/Desktop/HTML/DBMS_OMTB/dbms1.html

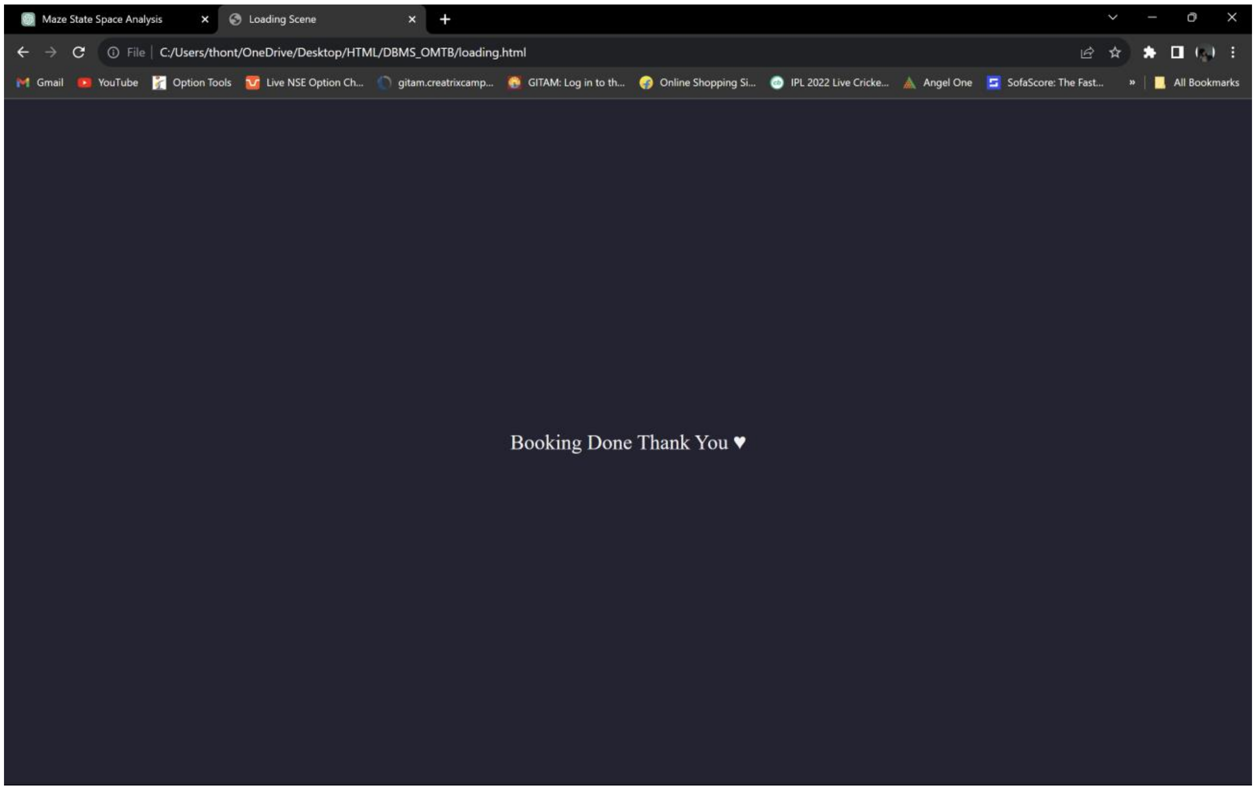


Pick a movie: RRR (240 rps)

You have selected 2 seats for a price of rps 300


Maze State Space Analysis
Loading Scene
File
C:/Users/thont/OneDrive/Desktop/HTML/DBMS_OMTB/loading.html
Gmail
YouTube
Option Tools
Live NSE Option Ch...
gitam.creatrixcamp...
GITAM: Log in to th...
Online Shopping Si...
IPL 2022 Live Cricke...
Angel One
SofaScore: The Fast...
All Bookmarks

Loading...



Booking Done Thank You ♥

Conclusion:

The Online Movie Ticket System (OMTS) is a modern solution that revolutionizes the way individuals purchase movie tickets. By providing a seamless and user-friendly interface for browsing movies, checking showtimes, seat selection, and secure online payments, OMTS significantly improves the overall convenience and efficiency of the movie ticket booking process. This digital innovation caters to the evolving needs and preferences of moviegoers, making it easier than ever to enjoy a cinematic experience.

In conclusion, the Online Movie Ticket System (OMTS) is a testament to the digital innovation that aims to simplify and enhance the movie ticket booking process. It offers a myriad of benefits to moviegoers, theaters, and the entire entertainment industry. As it becomes more widely adopted, OMTS is likely to play a pivotal role in shaping the future of movie ticket booking, ensuring that enjoying a cinematic experience is not only possible but also incredibly convenient.

References :

"HTML, CSS, and JavaScript." MDN Web Docs.
https://developer.mozilla.org/
• The project uses HTML, CSS, and JavaScript for building the user interface. The MDN Web Docs provide comprehensive guides and references for web development technologies.

"MySQL." MySQL Official Website.
https://www.mysql.com/
• MySQL is the database management system used for storing user data, medication information, and schedules. You can find resources and documentation on the official MySQL website.