

DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY KUDLU GATE
BANGALORE – 560068



A Mini Project Report On **“Movie Ticket Booking System”**

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

Name: Abdul Hafeez

USN: ENG21CS0004

Name: Bhagyalakshmi R.A

USN: ENG21CS0071

Name: Aditya Pandey

USN: ENG21CS0014

Name: Bhat Dhanvi

USN: ENG21CS0074

Under the Guidance of:
Prof. Rashmi Mothkur
Designation: Assistant Professor
V Semester, 2023

Course Code: 21CS3501 Course: Database Management System

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068



CERTIFICATE

This is to certify that Abdul Hafeez, Aditya Pandey, Bhagyalakshmi R.A, Bhat Dhanvi bearing USN ENG21CS0004, ENG21CS0014, ENG21CS0071, ENG21CS0074, have satisfactorily completed their Database Management System Mini Project Report as prescribed by the University for the Fifth semester B.Tech. Program in Computer Science & Engineering during the year 2023 at the School of Engineering, Dayananda Sagar University, Bangalore

Date: 19/12/2023

Faculty in Charge

Chairman

Department of Computer Science & Engineering

ACKNOWLEDGEMENT

The success and final outcome of this software requirement document required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Dr. GIRISHA G S, Professor and Chairman**, for providing us an opportunity and giving us all support and guidance, which made us complete the project duly. We are extremely thankful to him for providing such nice support and guidance, even though he has a busy schedule managing the departmental affairs.

We owe our deep gratitude to **our guide Prof. RASHMI MOTHKUR** for guiding us all along, till the completion of our project work by providing all the necessary information for the successful completion of our mini project work.

We are thankful to, and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Science Engineering department, which helped us in successfully completing our report. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

Student Names

- 1.Abdul Hafeez
- 2.Aditya Pandey
- 3.Bhagyalakshmi R.A
- 4.Bhat Dhanvi

ABSTRACT

The Movie Ticket Booking System is an online platform designed to facilitate the seamless booking of movie tickets. Users can access the system through a user-friendly interface, browse available movies, view show times, and see the availability of seats. However, the system is designed with the future aim of incorporating transaction capabilities. While currently focused on offering users information and enhancing their browsing experience, the ultimate goal is to enable seamless online ticket booking and transactions in the future. This approach allows for a phased development, starting with information dissemination and progressively integrating transactional features to meet evolving user needs.

Table of Contents

SL.NO	Page No
1.	Introduction
	1.1 Problem Statement
	1.2 Objectives of the project
2..	System Requirements
	2.1 Software and Hardware Requirements
3.	System Design
	3.1 ER diagram
4.	System Implementation
	4.1 Module Description
5.	Output Screen Shots
6.	Conclusion
	References

1. INTRODUCTION

1.1 PROBLEM STATEMENT

The traditional process of movie ticket booking poses challenges for users in terms of convenience and flexibility. Existing systems often lack a user-friendly interface for browsing movie details and face difficulties in facilitating online transactions. Moreover, the absence of a straightforward mechanism for users to cancel or delete bookings adds to the inconvenience. This results in a suboptimal user experience, potentially leading to decreased user engagement.

In light of these challenges, our project aims to provide a comprehensive solution by developing a Movie Ticket Booking System. This system will not only offer detailed movie information but also streamline the ticket booking process through an intuitive online interface. Furthermore, the system will empower users with the ability to easily cancel or delete bookings, addressing a crucial aspect of user flexibility. By addressing these pain points, our project seeks to enhance the overall movie-going experience and bring convenience to users engaging with the platform.

1.2 OBJECTIVES OF THE PROJECT

1.2.1 Booking Deletion Feature:

Implement a user-friendly feature that allows users to easily delete or cancel movie bookings if they are no longer needed, providing greater flexibility and control.

1.2.2 User Account Management:

Create a user account system to enable users to track their booking history, save preferences, and manage their profiles for a personalized experience.

1.2.3 Enhanced User Interface:

Design an intuitive and responsive interface that simplifies the movie browsing and booking process, ensuring a positive and efficient user experience.

2.SYSTEM REQUIREMENTS

2.1 SOFTWARE REQUIREMENTS

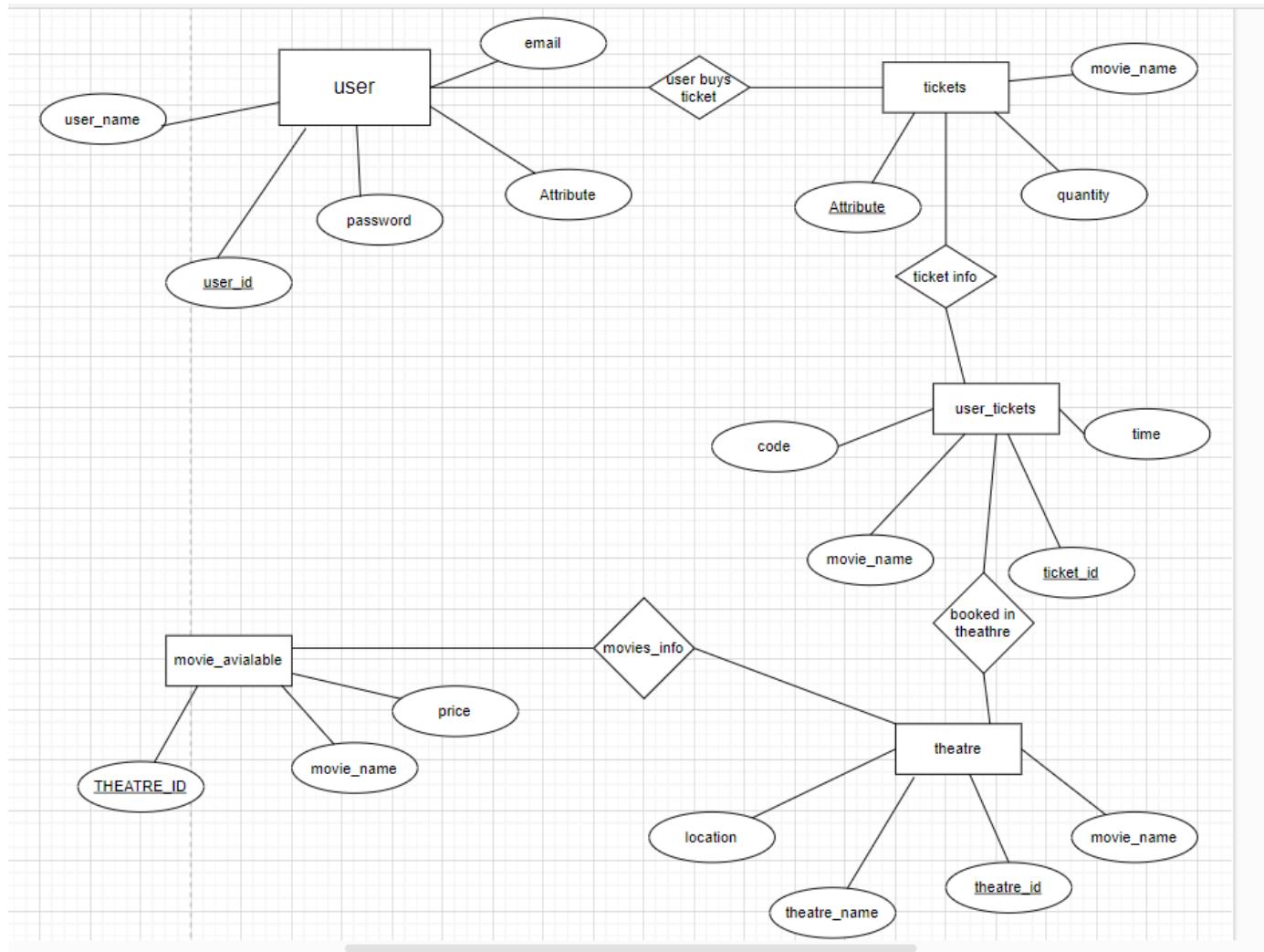
- 1 Operating System: Windows
- 2 Database Management System: MySQL
- 3 Server Side Scripting : Node.js
- 4 Framework : Express.js
- 5 Frontend Development: HTML, CSS, JavaScript
- 6 Integrated development environment (IDE) : Visual Studio

2.2 HARDWARE REQUIREMENTS

- 1 Processing Power: CPU
- 2 Memory : RAM
- 3 Secondary Storage : SSD
- 4 Web Browser: Google Chrome

3.SYSTEM DESIGN

3.1 ER DIAGRAM



4. SYSTEM IMPLEMENTATION

4.1 MYSQL CODE:

```
create database movie_ticket_management;
use movie_ticket_management;
```

```
create table user_(usr_id INT AUTO_INCREMENT, usr_name varchar(20), phn_no bigint,
usr_email varchar(10), password_ varchar(30), primary key(usr_id));
```

```
CREATE TABLE IF NOT EXISTS tickets (
    movie_id INT PRIMARY KEY,
    movie_name VARCHAR(255) NOT NULL,
    quantity INT NOT NULL
);
```

```
INSERT INTO tickets (movie_id, movie_name, quantity) VALUES
(1, 'Interstellar', 10),
(2, 'Baby"s Day Out', 8),
(3, 'The Avengers', 15),
(4, 'Movie 4', 5),
(5, 'Movie 5', 12);
```

```
CREATE TABLE IF NOT EXISTS user_tickets (
    ticket_id INT AUTO_INCREMENT PRIMARY KEY,
    movie_name VARCHAR(255) NOT NULL,
    time VARCHAR(20) NOT NULL,
    code VARCHAR(20) NOT NULL
);
select * from user_tickets;
delete from user_tickets;
```

```
CREATE TABLE THEATRE(THEATRE_ID INT PRIMARY KEY, THEATRE_NAME
VARCHAR(20), LOCATION VARCHAR(20), movie_name varchar(20));
drop table theatre;
insert into theatre value(1,"orion mall","bglr","inception"),
```

```
(2,"vishal mall","bglr","avengers"),
(3,"forum mall","bglr","baby day out"),
(4,"mantri mall","bglr","nightmare"),
(5,"mall of asia","bglr","interstellar");
```

```
CREATE TABLE MOVIE_AVAIL(THEATRE_ID INT, movie_name VARCHAR(20),
price int, FOREIGN KEY(THEATRE_ID) REFERENCES THEATRE(THEATRE_ID));
insert into MOVIE_AVAIL value(1,"interstellar",350),
(2,"avengers",780),
(3,"baby day out",450),
(4,"nightmare",380),
(5,"inception",680);
drop table MOVIE_AVAIL;
```

```
-- drop database movie_ticket_management;
```

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
'password';
flush privileges;
```

4.2 FRONTEND CODE

4.2.1 SIGN UP

HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sign up</title>
    <link rel="stylesheet" href="./sign_up.css">
    <link rel="icon" type="image/x-icon" href="..\User_Ticket_Page\Favicon.png">
</head>
<body>
    <!--  -->
    <div class="glass-card">
        <div class="head">
            <h1>SIGN UP</h1>
        </div>
        <br>
        <div>
            <form action="/sign_up" method="post">
                <div class="name_label">
                    <label for="name">ENTER FULL NAME</label>
                    <input type="text" name="name" id="name">
                </div>
                <div class="phone_label">
                    <label for="phone">ENTER PHONE NUMBER</label>
                    <input type="text" name="phone" id="phone">
                </div>
                <div class="email_label">
                    <label for="email">ENTER EMAIL</label>
                    <input type="text" name="email" id="email">
                </div>
                <br>
                <div class="password_label">
                    <label for="password">ENTER PASSWORD</label>
                </div>
            </form>
        </div>
    </div>
</body>
```

```
        <input type="password" name="password" id="password">
    </div>
    <div class="button">
        <button type="submit" ><a style="text-decoration: none;">sign
up</a></button>
    </div>

</form>
<p style="display: grid; justify-content: center; justify-items:
center;">don't have an account? <a href=".\\sign_in\\sign_in.html">Sign in</a>
</p>
</div>
</body>
</html>
```

CSS :

```
* {
    padding: 0;
    margin: 0;
    box-sizing: border-box;
    font-family: sans-serif;
}

body {
    background: url("./images/adrien-olichon-zyHJT71WwNU-unsplash.png")
no-repeat;
    background-position: center;
    background-size: cover;
    display: flex;
```

```
justify-content: center;  
  
align-items: center;  
}  
  
  
.head{  
  
font-family: Inter;  
  
font-size: 30px;  
  
font-style: italic;  
  
font-weight: 800;  
  
line-height: normal;  
  
background: linear-gradient(179deg, #48C8FF 0.93%, rgba(0, 255, 179, 0.59)  
114.37%);  
  
background-clip: text;  
  
-webkit-background-clip: text;  
  
-webkit-text-fill-color: transparent;  
  
margin-bottom: 60px;  
  
margin-top: 30px;  
}  
  
  
label{  
  
color: whitesmoke;  
  
font-size: 18px;
```

```
}

#email,#password{

    margin: 30px;

}

input{

    background-color: transparent;

    border: none;

    font-size: 18px;

    color: #00FCBF;

}

.email_label{

    border-bottom: 1px solid rgb(47, 255, 255);

    padding: 10px;

}

.password_label{

    border-bottom: 1px solid rgb(47, 255, 255);

    margin-top: 10px;

}
```

```
.name_label{  
    border-bottom: 1px solid rgb(47, 255, 255);  
    padding: 10px;  
}  
  
.phone_label{  
    border-bottom: 1px solid rgb(47, 255, 255);  
    padding: 10px;  
}  
  
}  
  
.glass-card {  
    padding: 10px;  
    margin-top: 100px;  
    background: white;  
    background: linear-gradient(  
        to right bottom,  
        rgba(58, 50, 50, 0.4),  
        rgba(66, 60, 60, 0.2)  
    );  
    backdrop-filter: blur(1rem);  
    width: 650px;
```

```
height: 850px;  
  
z-index: 1;  
  
display: flex;  
  
flex-direction: column;  
  
  
align-items: center;  
  
border-radius: 5rem;  
  
box-shadow: 0 0 1rem rgba(0, 0, 0, 0.2);  
}  
  
.button{  
  
border: solid 2px black;  
  
margin-top: 80px;  
  
margin-left: 180px;  
  
height: 100px;  
  
width: 180px;  
  
border-radius: 20px;  
  
display: flex;  
  
justify-content: center;  
  
border: 3px #00FCBF solid;  
}  
  
  
button{  
  
font-size: 30px;  
  
border: none;
```

```
background-color: transparent;

background: linear-gradient(179deg, #48C8FF 0.93%, rgba(0, 255, 179, 0.59)
114.37%);

background-clip: text;
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;

}

button:hover{

background: linear-gradient(179deg, #00fffb7 0.93%, rgba(0, 255, 179, 0.59)
114.37%);

background-clip: text;
-webkit-background-clip: text;
-webkit-text-fill-color: transparent;

}

p{

margin-top: 30px;
color: #00FCBF;
/* margin-right: 10px; */

}

a{
```

```
    color: #00FCBF;  
}
```

4.2.2 TICKETS

HTML :

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
    <meta charset="UTF-8">  
  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
    <title>User Movies</title>  
  
    <link rel="stylesheet" href="./ticket.css">  
  
    <link rel="icon" type="image/x-icon" href="..\User_Ticket_Page\Favicon.png">  
  
</head>  
  
<body>  
  
    <div class="container_nav">  
  
        <div class="mytickets">  
  
            <form action="/mytickets" method="get">  
  
                <button>My Tickets</button>  
  
            </form>  
  
        </div>  
  
        <div class="nav">
```

```
<button ><a style="text-decoration: none;" href="mYtickets.html">Home</a></button>

<button ><a style="text-decoration: none;" href="pricing.html">Movie Available</a></button>

<form action="/theatre" method="get">

    <button >Theatre</a></button>

</form>

</div>

</div>

<!-- <div class="content">

    <pre>          Wanna Have Fun? Check
                    Out These!
    </pre> -->

<!-- </div> -->

<div class="movie">

    <div class="m1">

        <form action="/interstellar" method="get">

            <button id="plus">+</button>

        </form>

        <!-- <button class="plus">add</button>

        </div>

        <div class="minus">

        </div> -->
```

```
<!-- <img class='image' src='installer.jpg' alt="Interstellar"
title="Interstellar" > -->

</div>

<div class="m2">

    <button id="plus2">
        +
    </button>

    <!-- <img class='image' src='BDO.png' alt="Baby's Day Out"
title="Baby's Day Out" > -->

</div>

<div class="m3">

    <button id="plus3">
        +
    </button>

    <!-- <img class='image' src='horror.jpg' alt="A Nightmare on ELM
Street" title="A Nightmare on ELM Street" > -->

</div>

<!-- <br> -->

<div class="m4">

    <button id="plus4">
        +
    </button>

    <!-- <img class='image' src='inception.jpg' alt="Inception"
title="Inception" > -->

</div>

<div class="m5">
```

```
<button id="plus5">  
    +  
</button>  
  
    <!-- <img class='image' src='peakpx.jpg' alt="The Avengers"  
title="The Avengers"> -->  
</div>  
  
</div>  
  
</body>  
</html>
```

CSS :

```
body {  
  
background: url("./Background.jpg") no-repeat;  
background-position: center;  
/* background-size: 1500px 1000px; */  
  
background-size: cover;  
opacity: 0.7;  
}  
  
.container_nav{  
display: flex;  
}  
  
.mytickets{
```

```
border: #00ffd0 3px solid;  
  
border-radius: 30px;  
  
font-size: 10px;  
  
margin: 20px;  
  
height: fit-content;  
  
width: 180px;  
  
padding: 5px;  
}  
  
  
a{  
  
color: #00ffd0;  
  
text-decoration: none;  
}  
  
  
button{  
  
border-radius: 20px;  
  
/* border: 2px solid #00ffd0 ; */  
  
color: #00ffcc8;  
  
background-color: transparent;  
  
font-size: 30px;  
}
```

```
a:hover{  
    color: rgb(0, 255, 85);  
}  
  
.nav{  
  
    height: 20px;  
    width: 100%;  
    margin: 20px;  
    display: flex;  
    flex-direction: row;  
    justify-content: space-around;  
  
}  
  
.content{  
  
    margin: 4px;  
}  
  
pre{  
    /* margin-top: 130px;  
    margin-bottom: 110px; */  
    color: #00fffd0;
```

```
font-size: 50px;  
font-family: 'Times New Roman', Times, serif;  
}  
  
.movie{  
  
padding: 20px;  
  
display: flex;  
  
flex-direction: row;  
  
justify-content: center;  
  
}  
  
.m1{  
  
display: flex;  
  
justify-content: center;  
  
align-items: last baseline;  
  
margin: 2px;  
  
height: 806px;  
  
width: 150px;  
  
background: url("./installer.jpg") no-repeat;  
  
background-size: cover;  
  
background-position: center;
```

```
}

.m2 {

    display: flex;

    justify-content: center;

    align-items: last baseline;

    margin: 2px;

    height: 806px;

    width: 150px;

    background: url("./BDO.png") no-repeat;

    background-size: cover;

    background-position: center;

}

.m3 {

    display: flex;

    justify-content: center;

    align-items: last baseline;

    margin: 2px;

    height: 806px;

    width: 150px;

    background: url("./peakpx.jpg") no-repeat;

    background-size: cover;
```

```
background-position: center;  
}  
  
.m4 {  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: last baseline;  
  
    margin: 2px;  
  
    height: 806px;  
  
    width: 150px;  
  
    background: url("./horror.jpg") no-repeat;  
  
    background-size: cover;  
  
    background-position: center;  
}  
  
.m5 {  
  
    display: flex;  
  
    justify-content: center;  
  
    align-items: last baseline;  
  
    margin: 2px;  
  
    height: 806px;  
  
    width: 150px;  
  
    background: url("./inception.jpg") no-repeat;  
  
    background-size: cover;
```

```
background-position: center;  
}  
  
.m1:hover {  
    /* display: block; */  
  
    transition: 2s;  
    width: 350px;  
    background-size: cover;  
}  
  
.m2:hover{  
    transition: 2s;  
    width: 350px;  
    background-size: cover;  
}  
  
.m3:hover{  
    transition: 2s;  
    width: 350px;  
    background-size: cover;  
}  
  
.m4:hover{  
    transition: 2s;
```

```
width: 350px;  
  
background-size: cover;  
}  
  
.m5:hover{  
  
transition: 2s;  
  
width: 350px;  
  
background-size: cover;  
}
```

JAVASCRIPT CODE

```
const express = require('express');  
  
const bodyParser = require('body-parser');  
  
const mysql = require('mysql');  
  
const cors = require('cors');  
  
const { exec } = require('child_process');  
  
  
const app = express();  
const PORT = 5500;  
  
  
app.use(cors());
```

```
const pool = mysql.createPool({  
  connectionLimit: 10,  
  host: 'localhost',  
  user: 'root',  
  password: 'password',  
  database: 'movie_ticket_management',  
  port: 3306,  
}) ;  
  
app.use(bodyParser.urlencoded({ extended: true }));  
  
app.use(bodyParser.json());  
  
app.use(express.static('view'))  
  
app.get('/', (req, res) => {  
  
  res.sendFile('/view/ticket.html', {root: __dirname});  
}) ;  
  
  
app.get('/updateTicketAndUser', (req, res) => {  
  console.log('Received request for /updateTicketAndUser');  
  
  pool.getConnection((err, connection) => {  
    console.log('Obtained database connection');  
  
    if (err) {  
      console.error('Error getting database connection', err);  
  
      return res.status(500).json({ success: false, message: 'Error getting database connection', error: err.message });  
    }  
  });  
});
```

```

}

const movieId = req.query.movieId;

// Check if tickets are available

connection.query(`SELECT * FROM tickets WHERE movie_id = ${movieId}`,
(error, results) => {

    if (error) {

        connection.release();

        console.error('Error fetching ticket information', error);

        return res.status(500).json({ success: false, message: 'Error
fetching ticket information', error: error.message });

    }

    if (results.length > 0 && results[0].quantity > 0) {

        const movieName = results[0].movie_name;

        // Subtract one ticket from the tickets table

        connection.query(`UPDATE tickets SET quantity = quantity - 1
WHERE movie_id = ${movieId}`, (error) => {

            if (error) {

                connection.release();

                console.error('Error updating ticket quantity', error);

                return res.status(500).json({ success: false, message:
'Error updating ticket quantity', error: error.message });

            }

        })
    }
})
}

```

```
// Insert a new row into the user_tickets table

const currentTime = new Date().toLocaleTimeString();

const ticketCode = `TKNO${results[0].quantity + 1}`;

connection.query(
  'INSERT INTO user_tickets (movie_name, time, code) VALUES
  (?, ?, ?)', [
    [movieName, currentTime, ticketCode],
    (error) => {
      connection.release();
      if (error) {
        console.error('Error updating user_tickets
table', error);
        return res.status(500).json({ success: false,
message: 'Error updating user_tickets table', error: error.message });
      }
      console.log('Ticket booked successfully');
      res.json({ success: true, message: 'Ticket booked
successfully' });
    }
  );
}

} else {
  connection.release();
}
```

```

        console.log('No tickets available');

        res.json({ success: false, message: 'No tickets available' });

    }

} );

}

);

app.listen(PORT, () => {

    console.log(`Server is running on http://localhost:${PORT}`);

});

```

4.2.3 MOVIES :

NIGHTMARE

```

<!DOCTYPE html>

<html>

<head>

    <meta charset='utf-8'>

    <meta http-equiv='X-UA-Compatible' content='IE=edge'>

    <title>A Nightmare on ELM Street</title>

    <meta name='viewport' content='width=device-width, initial-scale=1'>

    <link rel='stylesheet' type='text/css' media='screen' href='Movies.css'>

```

```
<link rel="icon" type="image/x-icon" href="..\User_Ticket_Page\Favicon.png">

</head>

<body style="background-image: url(nightmare_dark.png); background-size: cover;">

    <div class="mytickets">

        <button class="button"><a style="text-decoration: none; color: #00ffd0; font-size: 30px;" href=".ticket.html">Back</a></button>

    </div>

    <div style=" display: grid; justify-content: center;">

        <div class="div">

            <p style="font-size: 30px;">A Nightmare on ELM Street</p>

            Price: 380/-</p>

            <button id="plus" >BUY NOW</button>

        </div>

    </div>

    <script>

    </script>

</body>

</html>
```

INCEPTION

```

<!DOCTYPE html>

<html>
<head>

    <meta charset='utf-8'>

    <meta http-equiv='X-UA-Compatible' content='IE=edge'>

    <title>Inception</title>

    <meta name='viewport' content='width=device-width, initial-scale=1'>

    <link rel='stylesheet' type='text/css' media='screen' href='Movies.css'>

    <link rel="icon" type="image/x-icon" href="..\User_Ticket_Page\Favicon.png">

</head>

<body style="background-image: url(inception_dark.png); background-size: cover;">

    <div class="mytickets">

        <button class="button"><a style="text-decoration: none; color: #00ffff; font-size: 30px;" href=".//ticket.html">Back</a></button>

    </div>

    <div style=" display: grid; justify-content: center;">

        <div class="div">

            <p style="font-size: 30px;">Inception</p>

            <p>Price: 680/-</p>

            <button class="buyNow" onclick="print()">BUY NOW</button>

        </div>

    </div>

```

```

</div>

<script> function print() {
    window.alert("Ticket Booked!!");
}</script>

</body>
</html>

```

BABY'S DAY OUT

```

<!DOCTYPE html>

<html>

<head>

    <meta charset='utf-8'>
    <meta http-equiv='X-UA-Compatible' content='IE=edge'>
    <title>Baby's Day Out</title>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <link rel='stylesheet' type='text/css' media='screen' href='Movies.css'>
        <link rel="icon" type="image/x-icon"
        href=".\\User_Ticket_Page\\Favicon.png">
</head>

<body style="background-image: url(BOD_dark.png); background-size: cover;">

    <div class="mytickets">
        <button class="button"><a style="text-decoration: none; color: #00fffd0; font-size: 30px;" href=".\\ticket.html">Back</a></button>
    </div>
    <div style=" display: grid; justify-content: center;">

```

```

<div class="div">

    <p style="font-size: 30px;">Baby's Day Out</p>

    <p>Price: 450/-</p>

    <button class="buyNow" onclick="print()">BUY NOW</button>

</div>

</div>

<script> function print() {

    window.alert("Ticket Booked!!");

}</script>

</body>

</html>

```

INTERSTELLAR

```

<!DOCTYPE html>

<html>

<head>

    <meta charset='utf-8'>

    <meta http-equiv='X-UA-Compatible' content='IE=edge'>

    <title>Interstellar</title>

    <meta name='viewport' content='width=device-width, initial-scale=1'>

    <link rel='stylesheet' type='text/css' media='screen'
href='./Movies.css'>

```

```

        <link rel="icon" type="image/x-icon"
      href="..\User_Ticket_Page\Favicon.png">

</head>

<body style="background-image: url(Interstellar_dark.png); background-size:
cover;">

        <div class="mytickets">

            <button class="button"><a style="text-decoration: none; color:
#00fffd0; font-size: 30px;" href=".//ticket.html">Back</a></button>

        </div>

        <div style=" display: grid; justify-content: center;">

<div class="div">

            <p style="font-size: 30px;">Interstellar</p>

            Price: 350/-</p>

            <button id="plus" >BUY NOW</button>

        </div>

        </div>

<script> function bookTicket(movieId) {

fetch(`http://localhost:5500/updateTicketAndUser?movieId=${movieId}`)

    .then(response => response.json())

    .then(data => {

        if (data.success) {

            alert("Ticket booked successfully!");
        }
    })
}

```

```

        } else {

            alert('Error: ${data.message}');

        }

    } )

.catch(error => {

    console.error("Error:", error);

    alert("An error occurred while booking the ticket.");

}) ;

}

document.getElementById("plus").addEventListener("click", function() {

    bookTicket(1);

}) ;

</script>

</body>

</html>

```

AVENGERS

```

<!DOCTYPE html>

<html>

<head>

<meta charset='utf-8'>

<meta http-equiv='X-UA-Compatible' content='IE=edge'>

```

```
<title>Avengers</title>

<meta name='viewport' content='width=device-width, initial-scale=1'>

<link rel='stylesheet' type='text/css' media='screen' href='Movies.css'>

<link rel="icon" type="image/x-icon"
href=".\\User_Ticket_Page\\Favicon.png">

</head>

<body style="background-image: url(avengers_dark.png); background-size:
cover;">

<div class="mytickets">

    <button class="button"><a style="text-decoration: none; color:
#00ffd0; font-size: 30px;" href=".\\ticket.html">Back</a></button>

</div>

<div style=" display: grid; justify-content: center;">

<div class="div">

    <p style="font-size: 30px;">Avengers</p>

        Price: 780/-</p>

        <button class="buyNow" onclick="print()">BUY NOW</button>

</div>

</div>

<script> function print() {

    window.alert("Ticket Booked!!");

}</script>

</body>
```

```
</html>
```

4.2.3 PRICING

HTML :

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Movie_Available</title>

    <link rel="stylesheet" href="./pricing.css">

        <link rel="icon" type="image/x-icon"
    href="..\User_Ticket_Page\Favicon.png">

</head>

<body>

    <div class="container_nav">

        <div class="mytickets">

            <button class="button"><a style="text-decoration: none; color: #00ffd0; font-size: 30px;" href=".user.html">My Tickets</a></button>

        </div>

        <div class="nav">

            <button class="button"><a style="text-decoration: none; color: #00ffd0; font-size: 30px;" href="ticket.html">Home</a></button>

        </div>

    </div>

</body>
```

```
        <button class="button"><a style="text-decoration: none; color: #00fffd0; font-size: 30px;" href="pricing.html">Movie Available</a></button>

        <button class="button"><a style="text-decoration: none; color: #00fffd0; font-size: 30px;" href="About_us.html">About Us</a></button>

    </div>

</div>
```



```
<div id="main">

    <div class="Movies">

        <p>Interstellar</p>

        Price: 350/-</p>

        <button id="plus" >BUY NOW</button>

    </div>

    <div class="Movies" style="margin-left: 5%;"

        <p>Baby's Day Out</p>

        Price: 450/-</p>

        <button id="plus2" >BUY NOW</button>

    </div>

    <div class="Movies" style="margin-left: 5%;"

        <p>A Nightmare on ELM Street</p>
```

```

        <p>Price: 380/-</p>

        <button id="plus3" >BUY NOW</button>

    </div>

    <div class="Movies" style="margin-left: 10%;>

        <p>Avengers</p>

        <p>Price: 780/-</p>

        <button id="plus4" >BUY NOW</button>

    </div>

    <div class="Movies" style="margin-left: 15%;>

        <p>Inception</p>

        <p>Price: 680/-</p>

        <button id="plus5" >BUY NOW</button>

    </div>

</div>

<script>

    function bookTicket(movieId)  {

        fetch(`http://localhost:5500/updateTicketAndUser?movieId=${movieId}`)

            .then(response => response.json())

```

```
.then(data => {

    if (data.success) {

        alert("Ticket booked successfully!");

    } else {

        alert('Error: ${data.message}');

    }

}) 

.catch(error => {

    console.error("Error:", error);

    alert("An error occurred while booking the ticket.");

}) ;

}

document.getElementById("plus").addEventListener("click", function()

{

    bookTicket(1);

}) ;

document.getElementById("plus2").addEventListener("click", function() {

    bookTicket(2);

}) ;
```

```
        document.getElementById("plus3").addEventListener("click",
function()  {

    bookTicket(3);

} );

        document.getElementById("plus4").addEventListener("click",
function()  {

    bookTicket(4);

} );

        document.getElementById("plus5").addEventListener("click",
function()  {

    bookTicket(5);

} );

</script>

</body>

</html>
```

CSS

```
body{

background-image: url(Background_pricing.png);

background-size: cover;

}
```

```
.container_nav{  
    display: flex;  
}  
  
.nav{  
    height: 20px;  
    width: 100%;  
    margin: 20px;  
    display: flex;  
    flex-direction: row;  
    justify-content: space-around;  
}  
  
.mytickets{  
    border: #00ffd0 3px solid;  
    border-radius: 30px;  
    font-size: 10px;  
    margin: 20px;  
    height: fit-content;  
    width: 180px;  
    padding: 5px;  
}
```

```
.button{  
background-color: transparent;  
border-radius: 30px;  
}  
  
#main{  
width: 1250px;  
height: 1000px;  
margin-top: 2%;  
display: flex;  
flex-wrap: wrap;  
}  
  
.Movies{  
height: 380px;  
width: 350px;  
border-radius: 35px;  
border-color: #00ffd0;  
border-style: solid;  
color:#00ffd0;  
font-size: 20px;
```

```
}

p{
    display: grid;
    justify-content: center;
}

#plus{
    margin-left: 40%;
    border: #00ffd0 2px solid;
    border-radius: 15px;
    font-size: 15px;
    color: red;
    background-color: transparent;
    height: 40px;
    width: 60px;
}

#plus2{
    margin-left: 40%;
    border: #00ffd0 2px solid;
    border-radius: 15px;
    font-size: 15px;
```


BACKEND CODE

```
const express = require("express");
const mysql = require("mysql");
const bodyParser = require("body-parser");
const cors = require('cors');
```

```
const { exec } = require('child_process')

const app = express();
const port = 5500;
app.use(cors());

// Set up MySQL connection
const connection = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "movie_ticket_management",
}) ;

const pool = mysql.createPool({
  connectionLimit: 10,
  host: 'localhost',
  user: 'root',
  password: 'password',
  database: 'movie_ticket_management',
  port: 3306,
}) ;
```

```
// Connect to MySQL

connection.connect((err) => {

  if (err) {

    console.error("Error connecting to MySQL: " + err.stack);

    return;
  }

  console.log("Connected to MySQL as id " + connection.threadId);

}) ;

app.use(express.static('view'))

app.get("/sign_up", (req, res) => {

  res.sendFile(

    "./view/sign_up.html", {root: __dirname}

  );
}

);

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.post("/sign_up", (req, res) => {

  const name = req.body.name;

  const phn_no = req.body.phone;

  const usr_email = req.body.email;
```

```
const password = req.body.password;

// Perform database query to insert user information

const query = `INSERT INTO user_ (usr_name, phn_no,usr_email, password_) VALUES
(?, ?, ?, ?)`;

connection.query(
  query, [name, phn_no, usr_email, password], (err, result) => {
    if (err) {
      console.error("Error executing MySQL query: " + err.stack);
      res.status(500).send("Internal Server Error");
      return;
    }

    console.log("User added to the database");

    if(res.status(200)){
      res.sendFile('./view/ticket.html',{root:__dirname})
    }
  }
);

}) ;

app.get('/', (req, res) => {
  res.sendFile('/view/ticket.html',{root:__dirname});
}) ;
```

```
app.get('/pricing', (req, res) => {
    res.sendFile('/view/pricing.html', {root: __dirname});
}) ;

app.get('/interstellar', (req, res) => {
    res.sendFile('/view/intersellar.html', {root: __dirname});
}) ;

app.get('/updateTicketAndUser', (req, res) => {
    console.log('Received request for /updateTicketAndUser');

    pool.getConnection((err, connection) => {
        console.log('Obtained database connection');

        if (err) {
            console.error('Error getting database connection', err);
            return res.status(500).json({ success: false, message: 'Error getting database connection', error: err.message });
        }

        const movieId = req.query.movieId;

        // Check if tickets are available
    })
}) ;
```

```

    connection.query(`SELECT * FROM tickets WHERE movie_id = ${movieId}`,
(error, results) => {

    if (error) {

        connection.release();

        console.error('Error fetching ticket information', error);

        return res.status(500).json({ success: false, message: 'Error
fetching ticket information', error: error.message });

    }

    if (results.length > 0 && results[0].quantity > 0) {

        const movieName = results[0].movie_name;

        // Subtract one ticket from the tickets table

        connection.query(`UPDATE tickets SET quantity = quantity - 1 WHERE
movie_id = ${movieId}`, (error) => {

            if (error) {

                connection.release();

                console.error('Error updating ticket quantity', error);

                return res.status(500).json({ success: false, message:
'Error updating ticket quantity', error: error.message });

            }

            // Insert a new row into the user_tickets table

            const currentTime = new Date().toLocaleTimeString();

            const ticketCode = `TKNO${results[0].quantity + 1}`;


```

```
connection.query(`  
    'INSERT INTO user_tickets (movie_name, time, code) VALUES  
    (?, ?, ?)',  
    [movieName, currentTime, ticketCode],  
    (error) => {  
        connection.release();  
  
        if (error) {  
            console.error('Error updating user_tickets table',  
error);  
  
            return res.status(500).json({ success: false,  
message: 'Error updating user_tickets table', error: error.message });  
        }  
  
        console.log('Ticket booked successfully');  
  
        res.json({ success: true, message: 'Ticket booked  
successfully' });  
    }  
);  
});  
} else {  
    connection.release();  
  
    console.log('No tickets available');  
  
    res.json({ success: false, message: 'No tickets available' });  
}  
};  
});  
});
```

```
}) ;

// Serve the HTML page with the delete button

app.get('/mytickets', (req, res) => {

  // Fetch and display movies from the database

  const query = 'SELECT * FROM user_tickets';

  connection.query(query, (err, results) => {

    if (err) {

      console.error('Error executing MySQL query: ' + err.stack);

      res.status(500).send('Internal Server Error');

      return;
    }

    const html = `

      <!DOCTYPE html>

      <html lang="en">

        <head>

          <meta charset="UTF-8">

          <meta name="viewport" content="width=device-width, initial-scale=1.0">

          <title>Movies</title>

        </head>

        <body>

          <h1>Movies</h1>

          <ul>

            ${results.map(user_tickets => `

              <li>${user_tickets.movie_name}`

            )}

          </ul>

        </body>

      </html>
    `;
  })
});
```

```
<form action="/delete" method="post" style="display:inline;">

    <input type="hidden" name="id" value="\${user_tickets.ticket_id}">

    <button type="submit">Delete</button>

</form>

</li>`).join('')}

</ul>

</body>

</html>

`;

res.send(html);

}) ;

}) ;

app.post('/delete', (req, res) => {

const movieId = req.body.id;

const query = 'DELETE FROM user_tickets WHERE ticket_id = ?';

connection.query(query, [movieId], (err, result) => {

if (err) {

    console.error('Error executing MySQL query: ' + err.stack);

    res.status(500).send('Internal Server Error');

    return;
}

console.log('Movie deleted from the database');

```

```
    } ) ;  
  
} ) ;  
  
app.get('/theatre', (req, res) => {  
  
  const query = 'SELECT * FROM THEATRE';  
  
  connection.query(query, (err, results) => {  
  
    if (err) {  
  
      console.error('Error executing MySQL query: ' + err.stack);  
  
      res.status(500).send('Internal Server Error');  
  
      return;  
  
    }  
  
    const html = `  
  
      <!DOCTYPE html>  
  
      <html lang="en">  
  
        <head>  
  
          <meta charset="UTF-8">  
  
          <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  
          <title>Movies</title>  
  
        </head>  
  
        <body>  
  
          <h1>Movies</h1>  
  
          <ul>  
  
            ${results.map(THEATRE => `  
  
              <li>${THEATRE.THEATRE_ID} ${THEATRE.movie_name}  
${THEATRE.THEATRE_NAME} ${THEATRE.LOCATION}
```

```
        `).join('')}

    </ul>

</body>

</html>

`;

res.send(html);

}) ;

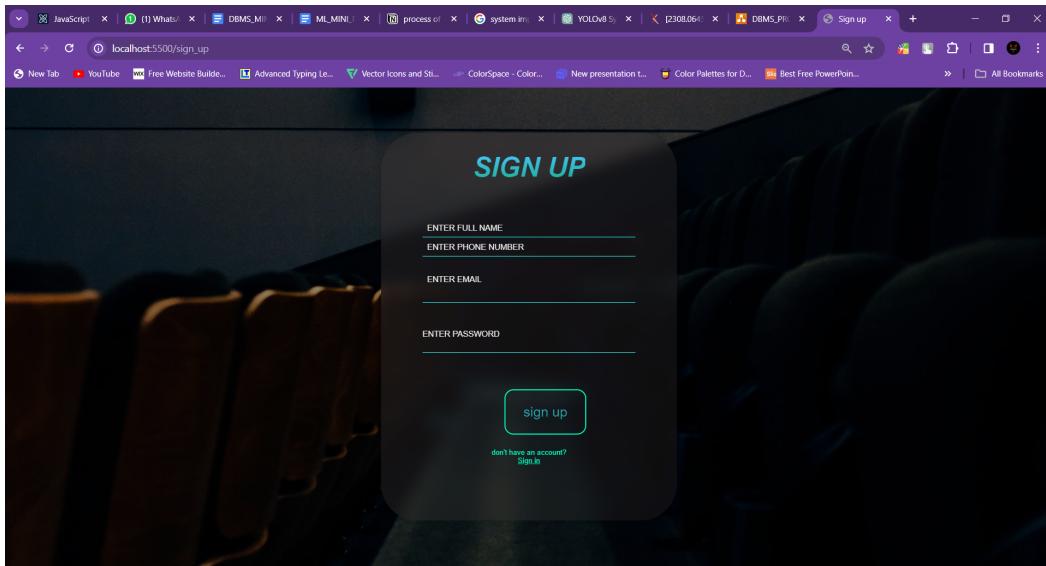
}) ;

app.listen(port, () => {

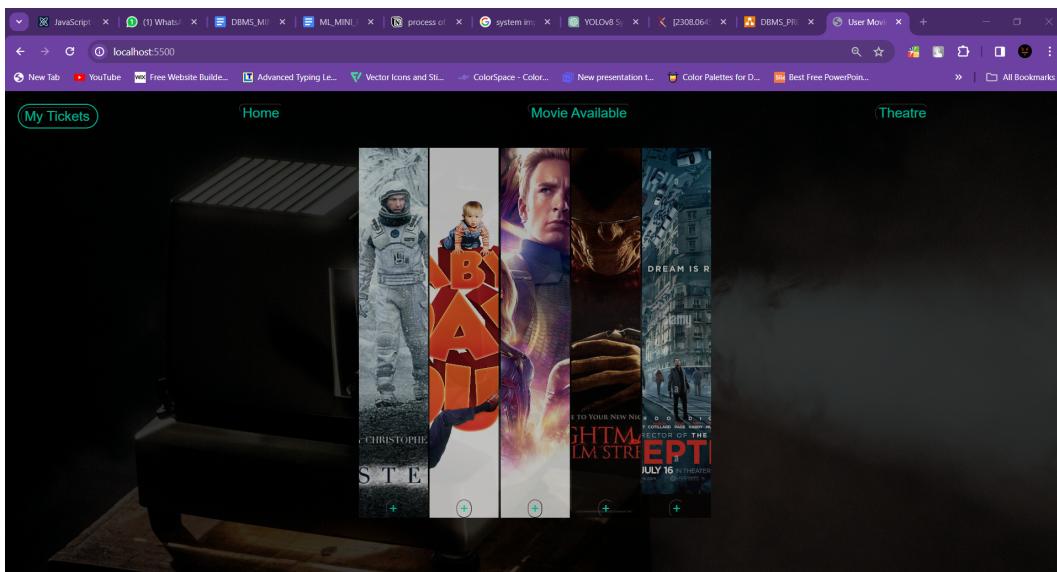
    console.log(`Server is running on port ${port}`);
}) ;
```

5. OUTPUTS

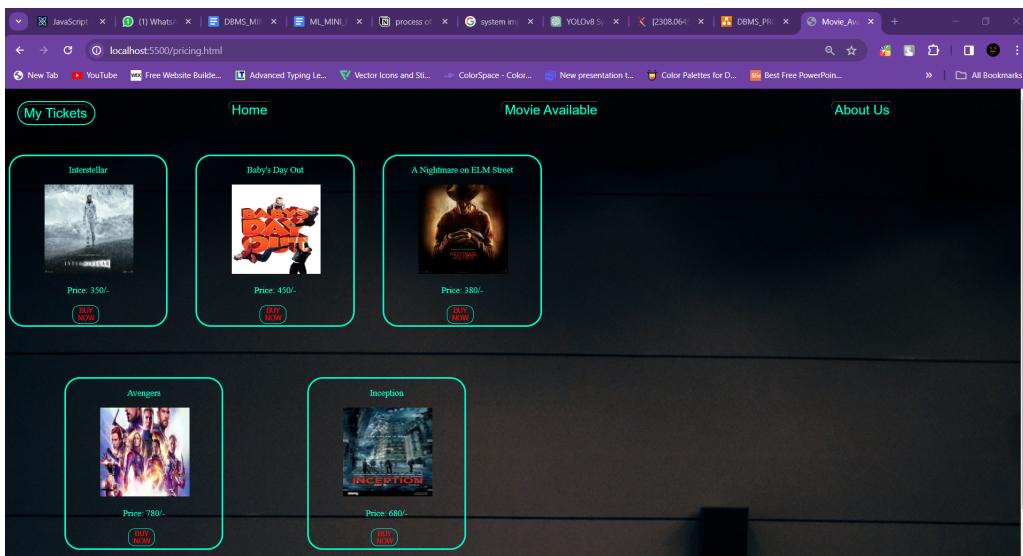
5.1 SIGN UP



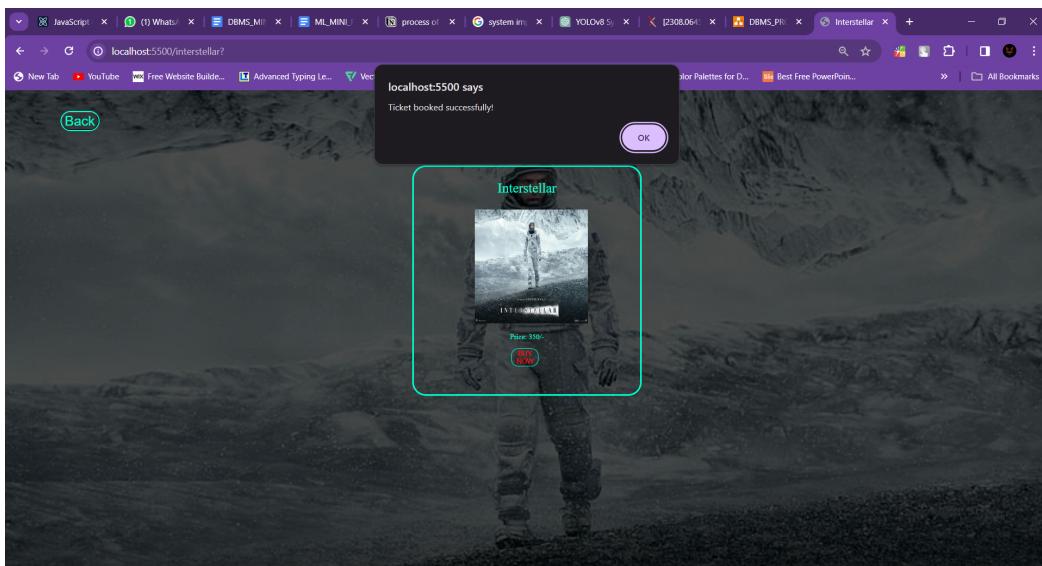
5.2 HOME



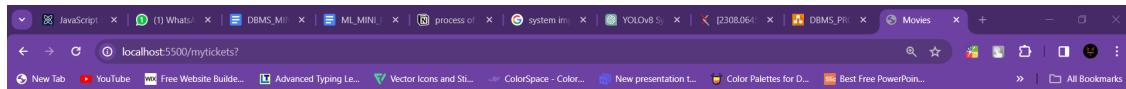
5.3 MOVIES AVAILABLE



4.4 MOVIE TICKET BOOKING:



5.5 BOOKED TICKETS



Movies

- Interstellar
- Interstellar

6.CONCLUSION

In conclusion, the development of a Movie Ticket Booking System involves careful consideration of software and hardware requirements to ensure an optimal user experience and efficient system performance. By focusing on enhancing user interaction through a streamlined interface, and incorporating online transaction capabilities in future, and providing features like booking deletion for user flexibility, the project aims to meet the evolving demands of the digital era. The hardware infrastructure, encompassing robust servers, database systems, and network components, must be configured to handle anticipated loads, with scalability in mind for future expansion. Overall, a well-designed Movie Ticket Booking System promises to offer users a seamless and secure platform for accessing movie information, making reservations, and enjoying the convenience of online ticketing.

REFERENCES

1. Rajouria, R., Yadav, V., Mishra, R., Mishra, R., & Jain, S. (2015). *Online Cinema Ticket Booking System*. *International Journal of Modern Engineering & Management Research*.
2. Shaik, A., Kishor, R. C. K., Koduri, S. S., & Anisha, P. R. (2014, October). *AMTS: Advanced Movie Ticketing System*. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies* .
3. AGBAEGBU, J., CALEB, C. V., & AYANGBEKUN, O. *Design and Implementation of a Movie Reservation System*.