ONLINE MOVIE TICKET RESERVATION SYSTEM

A MINI PROJECT REPORT

SUBMITTED BY

ARUN MOHAN 220701029

ANSH BHATIA 220701027

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE

(AUTONOMOUS) THANDALAM

CHENNAI-602105

2023-2024

BONAFIDE CERTIFICATE

Certified that this project report "**ONLINE MOVIE TICKET RESERVATION SYSTEM**" is the bonafide work of "**ARUN MOHAN (220701029), ANSH BHATIA (220701027)**"

who carried out the project work under my supervision

Submitted for the Practical Examination held on _____

SIGNATURE

Mr.N.Duraimurugan
Assistant Professor,
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous)
Thandalam, Chennai-602 105

INTERNAL EXAMINER   EXTERNAL EXAMINER

ABSTRACT

The online movie ticket reservation system is a sophisticated web-based platform designed to streamline the process of booking movie tickets for users. It offers a user-friendly interface where customers can easily browse through a comprehensive list of current movies, view detailed showtimes, and choose their preferred cinema locations. This system is accessible from any internet-enabled device, ensuring that users can make reservations conveniently from their homes, offices, or on the go.

One of the key features of the system is its real-time seat selection functionality. Users can view the seating layout of their chosen cinema hall and select their preferred seats based on availability. This interactive seating chart is updated in real time, ensuring that users have up-to-the-minute information about seat availability. Additionally, the system supports multiple payment gateways, allowing users to complete their transactions securely and efficiently.

For theater administrators, the online movie ticket reservation system provides robust tools for managing movie schedules, monitoring ticket sales, and generating comprehensive reports. The system's backend interface allows administrators to update movie listings, set showtimes, and adjust seat availability with ease. Detailed sales reports and analytics help theater management make informed decisions regarding promotions, pricing strategies, and capacity planning.

Overall, the online movie ticket reservation system significantly improves the movie-going experience for customers while enhancing operational efficiency for theaters. By offering a seamless and secure booking process, real-time updates, and valuable administrative tools, the system addresses the needs of both users and theater operators, leading to increased customer satisfaction and optimized theater management.

TABLE OF CONTENTS

CHAPTER 1.INTRODUCTION

1.1 INTRODUCTION :

An online movie ticket reservation system is an innovative digital platform designed to streamline and enhance the traditional process of purchasing movie tickets. By leveraging technology, this system offers a seamless, efficient, and user-friendly experience that allows users to book tickets from the comfort of their homes or on-the-go via mobile devices. The primary objective of this system is to eliminate the inconveniences associated with traditional ticket purchasing methods, such as standing in long queues at cinema counters or relying on third-party agents, thereby providing a more modern and convenient solution for moviegoers.

At the core of the online movie ticket reservation system is its robust search and filter functionality, which enables users to easily find movies based on various criteria including titles, genres, release dates, and locations. Users can browse a comprehensive catalog of movies, access detailed information about each film, including synopses, cast and crew details, ratings, and reviews, which aids in making informed decisions. The system also offers an interactive seating map feature that displays real-time seat availability, allowing users to select their preferred seats with clear visibility on different pricing tiers, ensuring a tailored and satisfactory booking experience.

The system's integration of secure payment gateways is a crucial aspect, ensuring that all transactions are conducted safely and user data is protected, thereby fostering trust and confidence among users. Multiple payment options such as credit/debit cards, net banking, mobile wallets, and UPI are supported to cater to diverse user preferences. Upon successful booking, users receive instant confirmation via email or SMS, which includes the booking details and a digital

Overall, an online movie ticket reservation system revolutionizes the movie-going experience by making it more accessible, efficient, and enjoyable. It caters to the evolving needs of modern consumers, offering a comprehensive solution that integrates convenience, security, and a rich set of features designed to enhance every aspect of booking movie tickets. As a result, moviegoers can look forward to a hassle-free and highly satisfying process that aligns with the fast-paced, technology-driven world of today.

1.2 OBJECTIVES:

• Convenience: The primary objective of the system is to provide users with a convenient way to book movie tickets without the need to visit physical ticket counters or rely on third-party agents. Users can access the system from anywhere with an internet connection, saving time and effort.

• Efficiency: By digitizing the ticket booking process, the system aims to streamline operations for both users and theater administrators. Users can quickly search for movies, select showtimes, choose seats, and make payments in a few simple steps. Similarly, administrators can manage movie listings, theater schedules, and bookings more efficiently through the system's backend interface.

• Accessibility: Another objective is to make the movie ticket booking process accessible to a wider audience. The system should be user-friendly and accessible to individuals of all ages and abilities, including those with disabilities. Features like screen reader compatibility, keyboard navigation, and clear interface design contribute to accessibility.

1.3 MODULES:
1..User Authentication and Authorization Module:
● Allows users to register, log in, and manage their profiles.
● Handles authentication and authorization to ensure secure access to the system.

2.Movie Listings Module:
● Displays a list of movies currently showing in theaters.
● Provides details such as movie title, description, genre, ratings, show timings, and available seats.

3.Booking Module:
● Enables users to select a movie, choose a showtime, and reserve seats. Manages the booking process, including seat selection, payment integration, and confirmation.
Seat Selection Module:
● Provides an interactive seat map for users to choose their seats.
Tracks seat availability in real-time and prevents double bookings.

4.Payment Gateway Integration Module:
● Integrates with a payment gateway to facilitate secure online payments.

- Supports various payment methods such as credit/debit cards, digital wallets, etc.

5. Booking Management Module:
- Allows users to view their booking history, cancel reservations, or modify booking details.
- Provides administrative tools for managing bookings, refunds, and resolving conflicts.

6. Admin Panel Module:
- Provides administrative functionalities for managing movies, theaters, showtimes, and user accounts.
- Allows administrators to monitor bookings, generate reports, and perform system maintenance tasks.

CHAPTER 2.SURVEY OF TECHNOLOGIES

2.1 SOFTWARE DESCRIPTION:

An **online** booking system is a software solution and reservation system that makes it simple for guests to book and pay for your tours and activities online. Some of these systems also include reporting software for tour operators and other user-friendly tools that help you improve efficiencies and boost bookings.

2.2 LANGUAGES:

2.2.1 PHP (Hypertext Pre-processor:

Role: Server-side scripting language.

Usage: PHP is used for backend development, handling server-side logic, processing user inputs, interacting with the database, and ensuring secure data transactions. Its ease of integration with HTML makes it an ideal choice for dynamic web applications.

Advantages:
- Open-source and widely supported.
- Efficient handling of server-side tasks.
- Seamless integration with various databases.

2.2.2 MySQL:

Role: Relational database management system.

Usage: MySQL is utilized for storing, retrieving, and managing all the data related to transactions, users, and system logs. It ensures data integrity and supports complex queries necessary for reporting and analytics.

Advantages:
- High performance and reliability.
- Robust security features.
- Scalability to handle growing amounts of data.

2.2.3 HTML (Hyper Text Markup Language)

Role: Standard markup language for creating web pages.

Usage: HTML is used to structure the content and layout of the web pages. It provides the basic framework of the system's user interface.

Advantages:
- Easy to learn and use.
- Universally supported by all web browsers.

2.2.4 CSS (Cascading Style Sheets)

Role: Style sheet language.

Usage: CSS is employed to style and layout the web pages, ensuring a visually

appealing and responsive design. It separates content from design, allowing for more flexible and maintainable code.

Advantages:
- Enhances the aesthetic appeal of web pages.
- Improves user experience through responsive design.
- Simplifies maintenance and updates.

2.2.5 Bootstrap

Role: Front-end framework.

Usage: Bootstrap is used to develop responsive and mobile-first web pages. It provides a collection of pre-designed components and utilities, speeding up the development process.

Advantages:
- Facilitates rapid development.
- Ensures consistent design across different devices.
- Extensive documentation and community support.

2.2.6 JavaScript

Role: Programming language for web development.

Usage: JavaScript is used for client-side scripting to create dynamic and interactive web pages. It controls the behaviour of different elements, enabling real-time updates and validations.

Advantages:
- Enhances interactivity and user engagement.
- Widely supported across all web browsers.
- Extensive ecosystem with numerous libraries and frameworks.

CHAPTER 3.REQUIREMENT AND ANALYSIS

3.1 REQUIREMENT SPECIFICATIONS:

User Requirements:
An online movie ticket reservation system should enable users to register, log in, and recover passwords easily. It should offer a robust search and filter function for movies by title, genre, and location, providing detailed movie and showtime information. Users must have access to an interactive seat selection feature with real-time availability and the ability to book multiple tickets in one transaction. The system should support various secure payment methods and send immediate booking confirmations via email or SMS. The interface should be user-friendly, accessible, and mobile-compatible.

System Requirements:
System requirements serve as a foundation for designing, implementing, and deploying a robust and reliable online movie ticket reservation system that meets the needs of stakeholders.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS:

Software Requirements:
- Operating System Windows 10
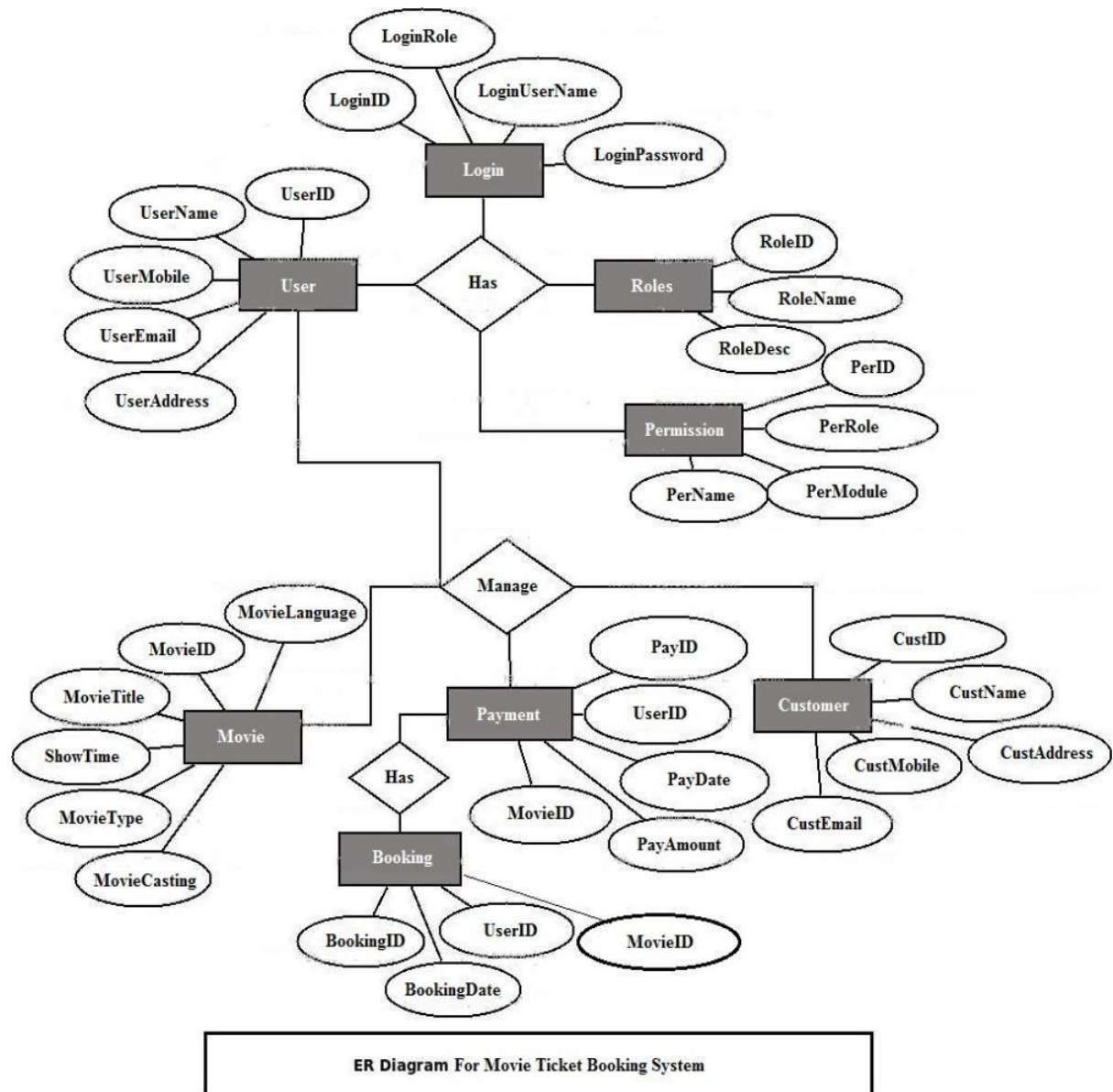- Front End HTML, CSS, JavaScript
- Back End PHP, MySQL
- search

Hardware Requirements:
- Desktop PC or a Laptop
- Printer
- Operating System – Windows 10
- Intel® Core TM i3-6006U CPU @ 2.00GHz
- 4.00 GB RAM
- 64-bit operating system, x64 based processor ● 1024 x 768 monitor resolution
- Keyboard and Mouse

3.3 ARCHITECTURE DIAGRAM:

3.4 ER DIAGRAM :
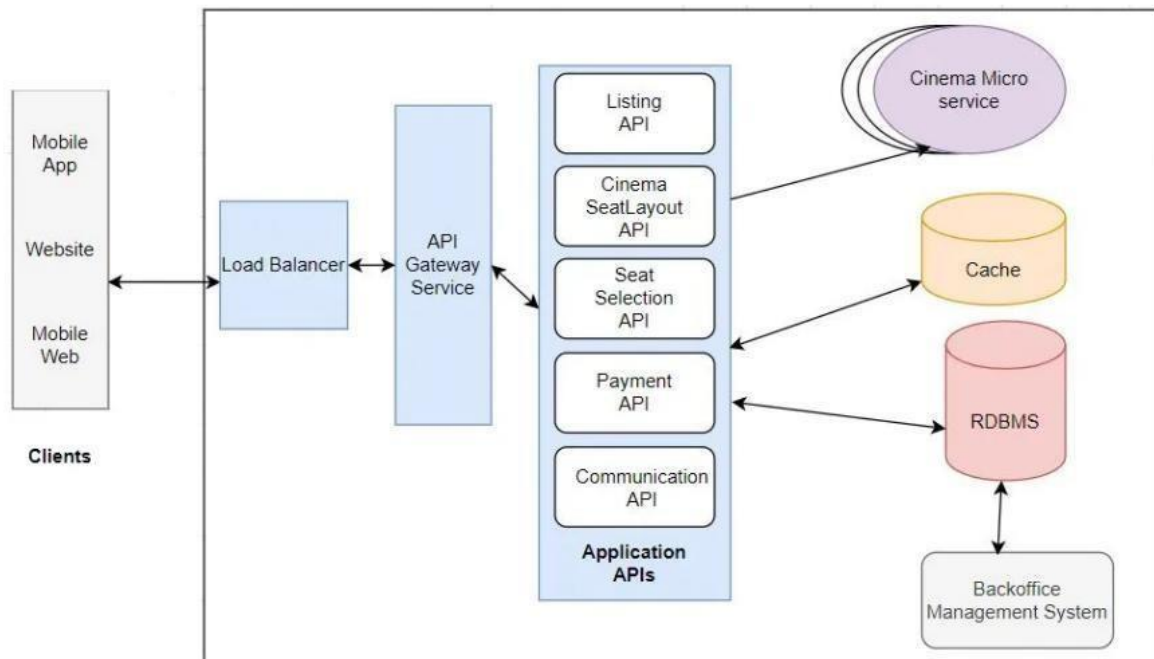
ER Diagram For Movie Ticket Booking System

CHAPTER 4.PROGRAM CODE

PROFILE CODING:

```php
<?php                              include('header.php');
if(!isset($_SESSION['user']))
{
header('location:login.php');
}
$qry2=mysqli_query($con,"select       *       from       tbl_movie       where
movie_id='".$_SESSION['movie']."'");
$movie=mysqli_fetch_array($qry2);
?>
<div class="content">
<div class="wrap">
<div class="content-top">
<div class="section group">
<div class="about span_1_of_2">
<h3 style="color:black;" class="text-center">BOOKING HISTORY</h3>
<?php include('msgbox.php');?>
<?php
$bk=mysqli_query($con,"select       *       from       tbl_bookings       where
user_id='".$_SESSION['user']."'");
if(mysqli_num_rows($bk))
{
?>
<table class="table table-bordered">
<thead>
 <th>Booking Id</th>
```

```php
<th>Movie</th>
<th>Theatre</th>
<th>Screen</th>
<th>Show</th>
<th>Seats</th>
<th>Amount</th>
<th></th>
</thead>
<tbody>
<?php
while($bkg=mysqli_fetch_array($bk))
$m=mysqli_query($con,"select * from tbl_movie where movie_id=(select movie_id from tbl_shows where s_id='".$bkg['show_id']."')");
$mov=mysqli_fetch_array($m)
$s=mysqli_query($con,"select * from tbl_screens where screen_id='".$bkg['screen_id']."'");
$srn=mysqli_fetch_array($s);
$tt=mysqli_query($con,"select * from tbl_theatre where id='".$bkg['t_id']."'");
$thr=mysqli_fetch_array($tt);
$st=mysqli_query($con,"select * from tbl_show_time where st_id=(select st_id from tbl_shows where s_id='".$bkg['show_id']."')");
$stm=mysqli_fetch_array($st);
?>
<tr>
<td>
<?php echo $bkg['ticket_id'];?>
</td>
<td>
```

```php
<?php echo $mov['movie_name'];?
</td>
<td>
<?php echo $thr['name'];?>
</td>
<td>
<?php echo $srn['screen_name'];?>
</td>
<td>
<?php echo $stm['name'];?>
</td>
<td>
<?php echo $bkg['no_seats'];?>
</td>
<td>
Rs. <?php echo $bkg['amount'];?>
</td>
<td>
<?php if($bkg['ticket_date']<date('Y-m-d'))
{
?>
<i class="glyphicon glyphicon-ok"></i>
<?php
}
else
{?>
<a href="cancel.php?id=<?php echo $bkg['book_id'];?>" style="text-decoration:none; color:red;">Cancel</a>
<?php
```

```php
 }
?>
</td>
</tr>
<?php
}
?></tbody>
</table>
<?php
}
else
{
?>
<h3    style="color:red;"    class="text-center">No    Previous    Bookings
Found!</h3>
<p>Once you start booking movie tickets with this account, you'll be able to see
all the booking history.</p>
<?php
}
?>
</div>
<?php include('movie_sidebar.php');?>
</div>
<div class="clear"></div>
</div>
</div>
</div>
<?php include('footer.php');?>
<script type="text/javascript">
```

```
$('#seats').change(function(){
var     charge=<?php     echo     $screen['charge'];?>;
amount=charge*$(this).val();
$('#amount').html("Rs "+amount);
$('#hm').val(amount);
});
</script>
```

REGISTRATION PAGE CODING:

```php
<?php include('header.php');?>
<link rel="stylesheet" href="validation/dist/css/bootstrapValidator.css"/>
<script                                        type="text/javascript"
src="validation/dist/js/bootstrapValidator.js"></script>
<!-- ================================================= -->
<?php
include('form.php');
$frm=new formBuilder;
?>
</div>
<div class="content">
<div class="wrap">
<div class="content-top" style="min-height:300px;padding:50px">
<div class="col-md-4 col-md-offset-4">
<div class="panel panel-default">
<div class="panel-heading">Register</div>
<div class="panel-body">
<form action="process_registration.php" method="post" id="form1">
<div class="form-group has-feedback">
<input name="name" type="text" size="25" placeholder="Name" class="form-
```

```
control"/>
<?php                                                                    $frm-
>validate("name",array("required","label"=>"Name","regexp"=>"name"));        //
Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-user form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input  name="age"  type="text"  size="25"  placeholder="Age"  class="form-
control"/>
<?php                                                                    $frm-
>validate("age",array("required","label"=>"Age","regexp"=>"age"));          //
Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-user form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<select name="gender" class="form-control">
<option value>Select Gender</option>
<option>Male</option>
<option>Female</option>
</select>
<?php      $frm->validate("gender",array("required","label"=>"Gender"));      //
Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-user form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input name="phone" type="text" size="25" placeholder="Mobile Number"
class="form-control"/
<?php               $frm->validate("phone",array("required","label"=>"Mobile
Number","regexp"=>"mobile")); // Validating form using form builder written
```

```
in form.php ?>
<span class="glyphicon glyphicon-phone form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input name="email" type="text" size="25" placeholder="Email" class="form-control"/>
<?php $frm->validate("email",array("required","label"=>"Email","email")); // Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-envelope form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input name="password" type="password" size="25" placeholder="Password" class="form-control" placeholder="Password" />
<?php $frm->validate("password",array("required","label"=>"Password","min"=>"7")); // Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="form-group has-feedback">
<input name="cpassword" type="password" size="25" placeholder="Password" class="form-control" placeholder="Password" />
<?php $frm->validate("cpassword",array("required","label"=>"Confirm Password","min"=>"7","identical"=>"password Password")); // Validating form using form builder written in form.php ?>
<span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="form-group">
<button type="submit" class="btn btn-primary">Continue</button>
```

```
</div>
</div>
</div>
</form>
</div>
</div>
<div class="clear"></div>
</div>
<?php include('footer.php');?>
</div>
<script>
<?php $frm->applyvalidations("form1");?>
</script>
```

LOGIN PAGE CODING:

```
[[[[[[[<?php include('header.php');?>
</div>
<div class="content">
<div class="wrap">
<div class="content-top" style="min-height:300px;padding:50px">
<div class="col-md-4 col-md-offset-4">
<div class="panel panel-default">
<div class="panel-heading">Login</div>
 <div class="panel-body">
<?php include('msgbox.php');?>
<p class="login-box-msg">Sign in to start your session</p>
<form action="process_login.php" method="post">
<div class="form-group has-feedback">
<input name="Email" type="text" size="25" placeholder="Email" class="form-
```

```html
control" placeholder="Email"/>
<span         class="glyphicon       glyphicon-envelope       form-control-
feedback"></span>
</div>
<div class="form-group has-feedback">
<input  name="Password"  type="password"  size="25"  placeholder="Password"
class="form-control" placeholder="Password" />
<span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="form-group">
<button type="submit" class="btn btn-primary">Login</button>
<p    class="login-box-msg"    style="padding-top:20px">New    Here?    <a
href="registration.php">Register</a></p>
</div>
</div>
</div>
</form>
</div>
</div>
<div class="clear"></div>
</div>
<?php include('footer.php');?>
</div>
```

LOGOUT PAGE CODING:

```php
<?php
session_start();
session_destroy();
header('location:index.php');
```

```
?>
```

COMPLETE PAYMENT CODING:

```php
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<?php
session_start();
if(!isset($_SESSION['user']))
{
header('location:login.php');
}
include('config.php');
extract($_POST);

//OTP          Code
if($otp=="123456")
{
$bookid="BKID".rand(1000000,9999999);
mysqli_query($con,"INSERT into tbl_bookings values(NULL,'$bookid','".$_SESSION['theatre']."','".$_SESSION['user']."','".$_SESSION['show']."','".$_SESSION['screen']."','".$_SESSION['seats']."','".$_SESSION['amount']."','".$_SESSION['date']."',CURDATE(),'1')");
$_SESSION['success']="Bookings Done!";
}
else
{
$_SESSION['error']="Payment Failed";
}
?>
```

```
<body><table align='center'><tr><td><STRONG>Transaction is being
processed,</STRONG></td></tr><tr><td><font color='blue'>Please Wait <i
class="fa fa-spinner fa-pulse fa-fw"></i>
<span class="sr-only"></font></td></tr><tr><td>(Do not 'RELOAD' this page
or 'CLOSE' this page)</td></tr></table><h2>
<script>
setTimeout(function(){ window.location="profile.php"; }, 3000);
</script>
```

PROCESS BOOKING CODING:

```php
<?php                               include('header.php');
if(!isset($_SESSION['user']))
{
header('location:login.php');
}?>
<link rel="stylesheet" href="validation/dist/css/bootstrapValidator.css"/>


<script                                             type="text/javascript"
src="validation/dist/js/bootstrapValidator.js"></script>
<!-- ================================================== -->
<?php
include('form.php');
$frm=new formBuilder;
?>
</div>
<div class="content">
<div class="wrap">
<div class="content-top">
<h3>Payment</h3>
```

```html
<form action="bank.php" method="post" id="form1">
<div class="col-md-4 col-md-offset-4" style="margin-bottom:50px">
<div class="form-group">
<label class="control-label">Name on Card</label>
<input type="text" class="form-control" name="name">
</div>
<div class="form-group">
<label class="control-label">Card Number</label>
<input type="text" class="form-control" name="number" required title="Enter 16 digit card number">

</div>
<div class="form-group">
<label class="control-label">Expiration date</label>
<input type="date" class="form-control" name="date">
</div>
<div class="form-group">
<label class="control-label">CVV</label>
<input type="text" class="form-control" name="cvv">
</div>
<div class="form-group">
<button class="btn btn-success">Make Payment</button>
</form>
</div>
</div>
</div>

<div class="clear"></div>
```

```php
</div>
<?php include('footer.php');?>
</div>
<?php session_start();
extract($_POST);
include('config.php');
$_SESSION['screen']=$screen;
$_SESSION['seats']=$seats;
$_SESSION['amount']=$amount;
$_SESSION['date']=$date;
header('location:bank.php');
?>
```

```javascript
<script>
$(document).ready(function() {
$('#form1').bootstrapValidator({
fields: {
name:
{      verbose:
false,
validators: {notEmpty: {
message: 'The Name is required and can\'t be empty'
},regexp: {
regexp: /^[a-zA-Z ]+$/,
message: 'The Name can only consist of alphabets'
} } },
number:
{      verbose:
false,
validators:
```

```
},stringLength:

{min: 16,

max: 16,

message: 'The Card Number must 16 characters long'

},regexp: {

regexp: /^[0-9 ]+$/,

message: 'Enter a valid Card Number'

} } },

date:

{       verbose:

false,

validators: {notEmpty: {

message: 'The Expire Date is required and can\'t be empty'

} } },

cvv:

{       verbose:

false,

validators: {notEmpty: {

message: 'The cvv is required and can\'t be empty'

},stringLength:

{min: 3,

max: 3,

message: 'The cvv must 3 characters long'

},regexp: {

regexp:     /^[0-9     ]+$/,

message: 'Enter a valid cvv'

} } }}

});

</script>
```

PROCESS LOGIN CODING:

```php
<?php
include('config.php');
session_start();
$email = $_POST["Email"];
$pass = $_POST["Password"];
$qry=mysqli_query($con,"select * from tbl_login where username='$email' and password='$pass'");
if(mysqli_num_rows($qry))
{
$usr=mysqli_fetch_array($qry);
if($usr['user_type']==2)
{
$_SESSION['user']=$usr['user_id'];
if(isset($_SESSION['show']))
{
header('location:booking.php');
}
else
{
header('location:index.php');
}
}
else
{
$_SESSION['error']="Login Failed!";
header("location:login.php");
}
```

```php
}
else
{
$_SESSION['error']="Login Failed!";
header("location:login.php");
}
?>
```

CHAPTER 5.RESULT AND DISCUSSION

5.1 LOGIN PAGE:
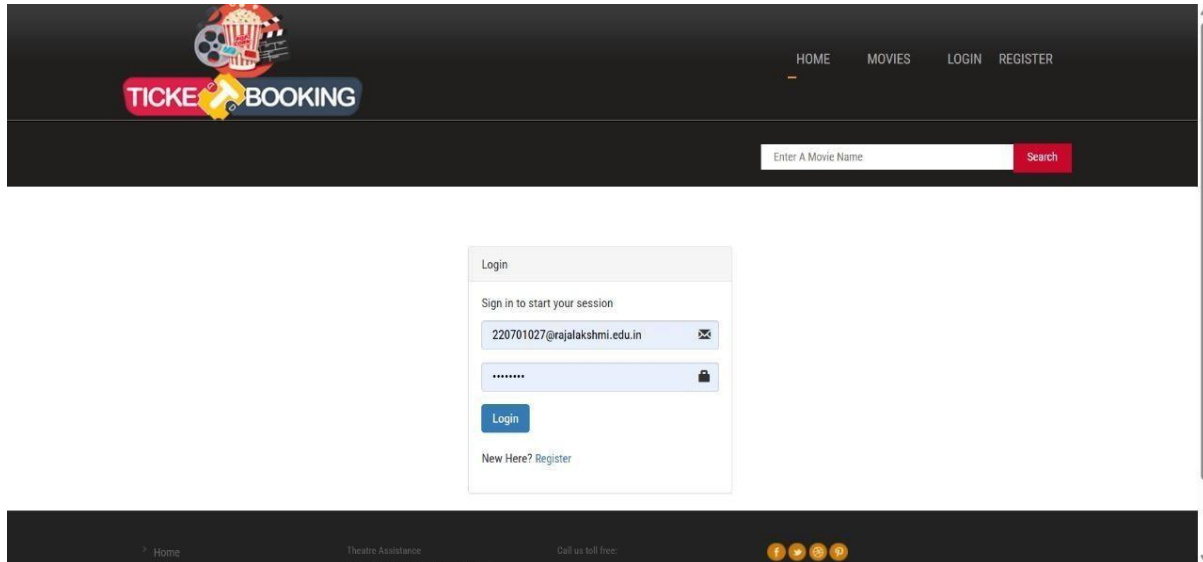
Figure 5.1

5.2 NOW SHOWING PAGE:
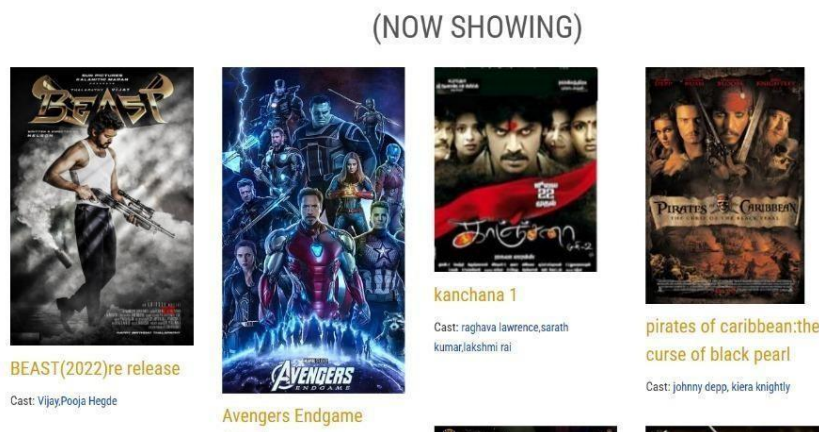
Figure 5.2

5.3 HOME PAGE:
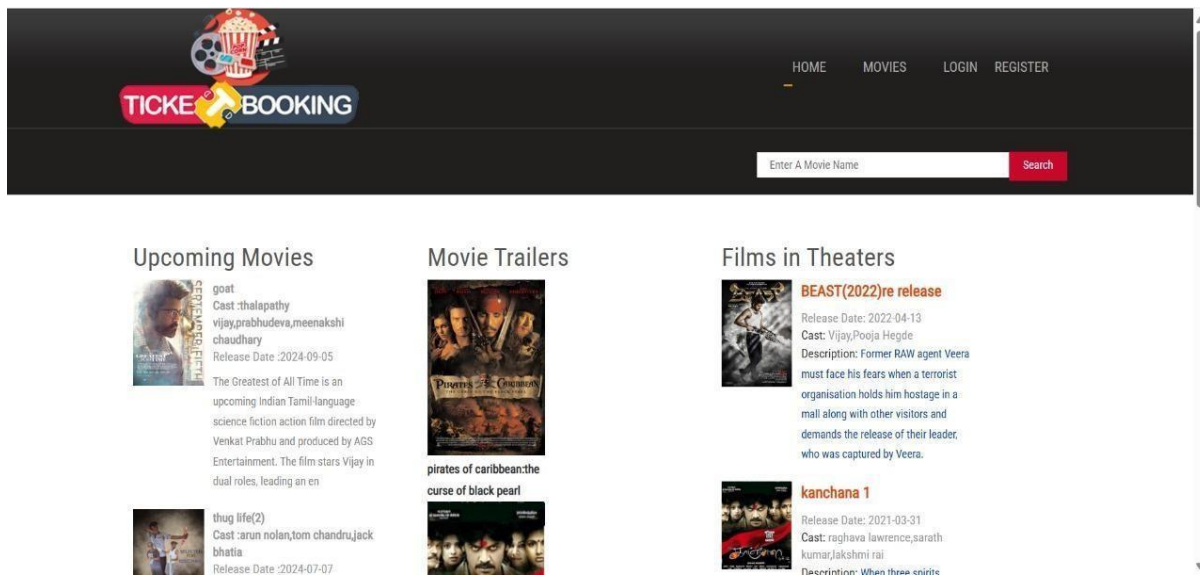
Figure 5.3

5.4 PAYMENT PAGE:

Figure 5.4

5.5 BOOKING HISTORY PAGE:

Figure 5.5

CHAPTER 6.CONCLUSION

6.1 FUTURE SCOPE:

This web application involves almost all the features of the online movie ticket booking. The future implementation will be online help for the customers and chatting with website adminstrator

6.2 CONCLUSION:

The online movie ticket reservation system has revolutionized the way audiences engage with cinema. By providing a convenient, efficient, and user- friendly platform for booking tickets, it enhances the overall movie-going experience. This system offers numerous benefits, including the ease of booking from anywhere, real-time updates on showtimes and seat availability, secure payment options, and valuable data analytics for theaters. Despite challenges such as technical issues and security concerns, continuous improvements and the integration of advanced technologies can further elevate the system's effectiveness. Ultimately, the online movie ticket reservation system stands as a testament to how digital solutions can streamline processes and meet the evolving needs of consumers in the entertainment industry.

   The use of PHP and MySQL provides a solid foundation for secure server-side processing and robust database management. HTML, CSS, and Bootstrap ensure a responsive and aesthetically pleasing user interface, while JavaScript enable real-time data processing and dynamic user interactions. The implementation of modal dialogs further enhances the system's usability by offering seamless and focused transaction handling.

Through detailed functional requirements, the system supports essential operations such as user authentication, transaction management, service provisioning, real-time data processing, reporting, and user management. Additionally, the non-functional requirements ensure the system's performance, usability, security, scalability, reliability, and maintainability.
This web application involves almost all the features of the online movie ticket booking. The future implementation will be online help for the customers and chatting with website adminstrator

CHAPTER 8.REFERENCES

To develop the online movie ticket reservation system using various resources and documentation were consulted. Below is a list of references that were instrumental in the development process:

YOUTUBE:

https://youtu.be/zCwnzcefaSo?si=X_0lUSyolhM0QC6g

GITHUB:

Movie Ticket Booking System in PHP with Source Code - CodeAstro

TICKETNEW:

Movies - Online Movie Ticket Booking & Get Cashback/Offers at PaytmTicket