VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi: 590018

Database Management Systems Mini Project report on

"Online Movie Ticket Booking"

Submitted in partial fulfilment of the requirement for the award of Degree of

BACHELOR OF ENGINEERING
IN
ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

By

Bhoomi Kaushik **1AY20AI012**
Amruta Maruti Khangaonkar **1AY20AI007**

Under the guidance of

Mr. Jovin Deglus

DEPARTMENT OF ARTIFICIAL INTELLIGENCE&MACHINE LEARNING ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

2022-2023

ACHARYA INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi) Soladevanahalli, Bangalore – 560090

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

CERTIFICATE

Certified that the Database Management Systems mini project entitled **ONLINE MOVIE TICKET BOOKING** is a bonafide work carried out by **Bhoomi Kaushik (1AY20AI012) & Amruta Maruti Khangaonkar (1AY20AI007)** of Fifth semester in partial fulfilment for the award of degree of **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University**, **Belagavi**, during the year **2022-2023.** It is certified that all corrections/ suggestions indicated for internal assessments have been incorporated in the Report deposited in the departmental library. The Mini Project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the **Bachelor of Engineering Degree**.

Signature of Coordinator **Signature of H.O.D**

Name of the examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

I express my gratitude to our institution and management for providing us with good infrastructure, laboratory, facilities and inspiring staff whose gratitude was of immense help in completion of this mini-project successfully.

I am grateful to the institute Acharya Institute of Technology and management with its ideas and inspiration for having provided us with the good infrastructure, laboratory, facilities and inspiring staff which has made this seminar report complete successfully.

I would like to express my sincere gratitude to **Dr. Rajath Hegde M M**, Principal, AIT for all the facilities that he has extended throughout my work.

I heartily thank and express my sincere gratitude to **Dr. Vijayashekhar S Sankannavar**, Associate Professor and Head, Dept. of AIML, AIT for his valuable support and a constant source of enthusiastic inspiration to steer us forward.

I would like to express my sincere gratitude to the Mini-project Coordinator **Mr. Jovin Deglus** Assistant Professor, Dept. of AIML, AIT for their valuable guidance and support.

Finally, I would like to express my sincere gratitude to my parents, all teaching and non-teaching faculty members and friends for their moral support, encouragement and help throughout the completion of the mini-project.

Bhoomi Kaushik 1AY20AI012

Amruta Maruti Khangaonkar 1AY20AI007

ABSTRACT

Online Movie Ticket Booking System is a PHP-based project implemented for a movie hall as the sole purpose of the booking system. This software has been developed using PHP, HTML, CSS, JavaScript and MySQL. The purpose of this website where users can log in to the system and book movie tickets online easily.

Online Movie Ticket Booking System can help users make booking movie tickets easy, flexible, and efficient. With this system, people no longer need to wait in long queues at the counters at the theatre to book their tickets. Instead, they can easily sit at home and book the required tickets online. With such software, the user can book tickets from anywhere and anytime with the internet's help. This system aims to allow customers to book movie tickets online and gather information about the movies and theatres without visiting the movie halls.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This **"Online Movie Ticket Booking System"** project aims to implement a movie ticket booking system using a web front end and a MySQL backend. It is a PHP-based project implemented for a movie hall as the sole purpose of the booking system. This software has been developed using PHP, HTML, CSS, and MySQL. The purpose of this website where users can log in to the system and book movie tickets online easily.

Here you will get a sample of 7 movies that are up in the project for booking as a sample. Each film on the website has its page with movie details and an option to make a booking. The website has a feature to book tickets in multiple theatres with multiple show timings that are provided. Functionalities like User login, Ticket Booking, and more are also supported. All system data is stored safely on tables in a MySQL server database. Users can book tickets for movies in theatres. Users can select their favourite film from their favourite theatre page.

ABOUT THE PROJECT:

Home page: After entering into the system home page will appear, In home page there is a user register option, admin and user signup options .And there are steps to know how this portal works

Admin page: As we login to the admin page we can manage all details like managing the seats, adding theatre, adding movie, adding slots etc.

User page: As we login to the user page we can see the user details of movies and select the movie, theatre, slot for booking the seats.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

Requirements specification is a specification of software requirements and hardware requirements required to do the project.

2.1 Hardware Requirements Specification:

Hardware Requirements are the hardware resources that are need to do the project work. These resources are a computer resource provides functions and services to do the project. Hardware resources required for our project are shown below.

- Processor: Intel i3 core and Above
- RAM: 1GB and Above
- Hard disk: Minimum 10 GB
- Keyboard: QWERTY

2.2 Software Requirements Specification:

Software Requirements are the software resources that are need to do the project work. These resources are installed on a computer in order to provide functions, services, hardware accessing capabilities to do the project.

In our project we used the following software resources.

- **Operating System** :Windows 7 and Above
- **Frontend** :HTML5, SASS 7.0.1, JavaScript ES2015
- **Backend** : PHP v7.4, MySQL v.8.0.26

2.3 Functional Requirements:

Functional requirements specify a function that system or a system component must be able to perform. It can be documented in various ways.

- Sign Up: The user should be able to set up a new account.

- Sign In: All the users and admin should be able to log onto the system with a username and password for each profile.
- Select Movie: Users can select their favourite movie that they want to book their tickets.
- Book Ticket: Users should now have to book movie tickets.
- Select Date: Users have to select the date of the movie.
- Select Showtime: They can select suitable showtime according to their choice
- Select Theatre: The users can easily select the theatre/cinema they want to book
- Request Ticket: The customers will now be able to verify the seats and other details before payment.
- Confirm Ticket: The User have successfully booked their movie ticket.

2.4 Non-Functional Requirements:

- Reliability: Database updating should follow transaction processing to avoid data inconsistency.
- Availability: The project will be deployed on a public shared server so it will be available all the time and will be accessible anywhere of the world using internet.
- Security: We have implemented a lot of security mechanism to avoid to hack the system by outer world.
- Maintainability: It is very easy to maintain the system. The system has been developed on PHP so anyone who has the knowledge of PHP, can easily maintain the system.
- Portability: Yes this system is portable and we can switch the servers very easily.
- Browser Compatibility: The project being web based required compatibility with at least the popular web browsers. Microsoft windows XP and above, Linux and Macintosh being the current popular operating system and Microsoft Internet Explorer, Mozilla Firefox, Opera, Safari and Google Chrome being the currently popular web browsers.

CHAPTER 3

TOOLS AND TECHNOLOGIES USED

3.1 PHP:

PHP known as “Hypertext Preprocessor”. It is a server-side scripting language usually written in an HTML context. Unlike an ordinary HTML page, a PHP script is not sent directly to a client by the server; instead, it is parsed by the PHP binary or module, which is server-side installed. HTML elements in the script are left alone, but PHP code is interpreted and executed. PHP code in a script can query databases, create images, read and write files, talk to remote servers – the possibilities is endless. The output from PHP code is combined with the HTML in the script and the result sent to the user’s web-browser, therefore it can never tell the user whether the web-server uses PHP or not, because the entire browser sees is HTML.

PHP’s support for Apache and MySQL .Apache is now the most-used web-server in the world, and PHP can be compiled as an Apache module. MySQL is a powerful free SQL database, and PHP provides a comprehensive set of functions for working with it.

PHP supports an extensive list of databases and web-servers.

Basic PHP Syntax :

A PHP scripting block always starts with <?php and ends with ?>. A PHP scripting block can be placed anywhere in the document.

A PHP file normally contains HTML tags, just like an HTML file, and some PHP scripting code.

3.2 HTML:

HTML or Hyper Text Markup Language is the standard markup language used to create web pages.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets(like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although

some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language.

3.3 CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a markup language.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content.

3.4 JAVASCRIPT:

JavaScript (JS) is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed. It is also being used in server-side network programming (with Node.js), game development and the creation of desktop and mobile applications.

3.5 BOOTSTRAP:

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages .The primary purpose of adding it to a web project is to apply Bootstrap's choices of colour, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container.

3.6 MYSQL

MYSQL is the language used to manipulate relational databases. It is tied closely with the relational model. It is issued for the purpose of data definition and data manipulation. Program runs as a server providing multi-user access to a number of databases. MySQL is a multithreaded, multi-user SQL database management system (DBMS). It includes facilities to add, modify or delete data from the database, ask questions (or queries) about the data stored in the database and produce reports summarizing selected contents.

CHAPTER 4:

SYSTEM DESIGN

4.1 Data Flow Diagram

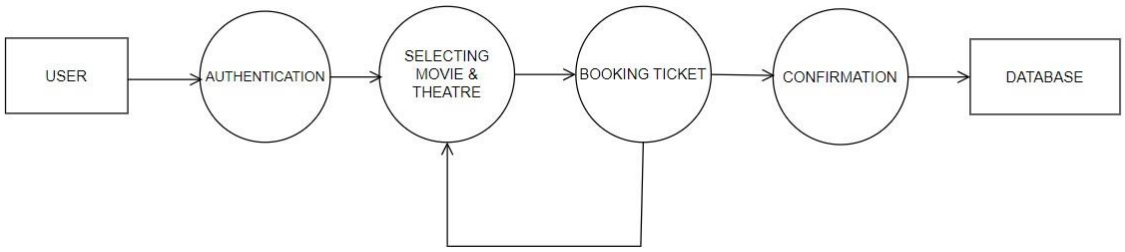


Figure: 4.1 Data Flow Diagram

4.2 Use Case Diagram :

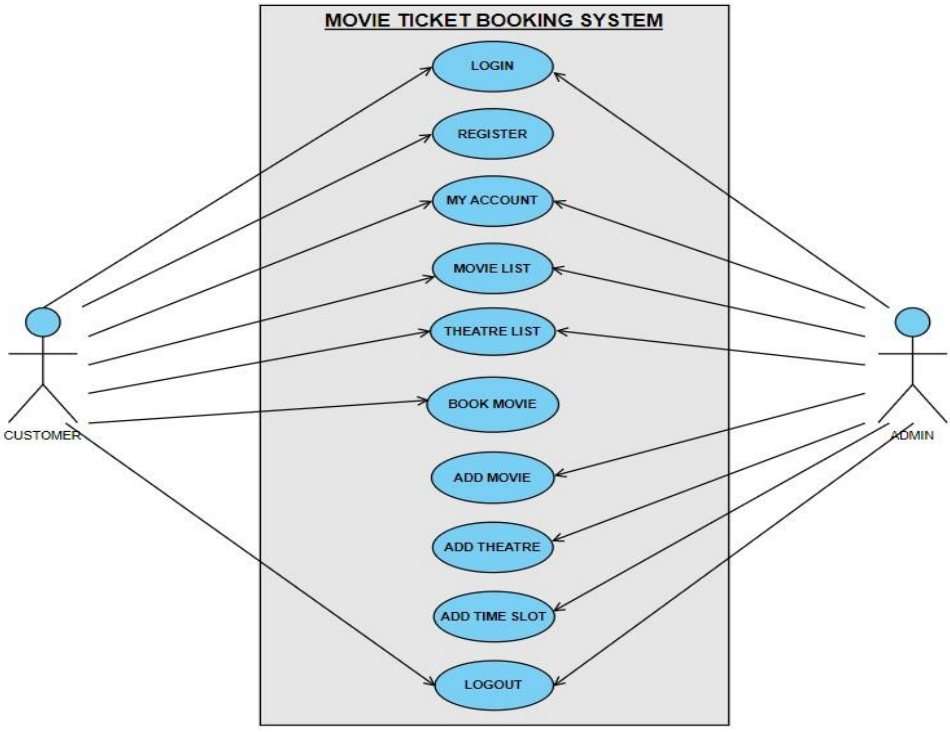


Figure: 4.2 Use Case Diagram

4.3 CLASS DIAGRAM

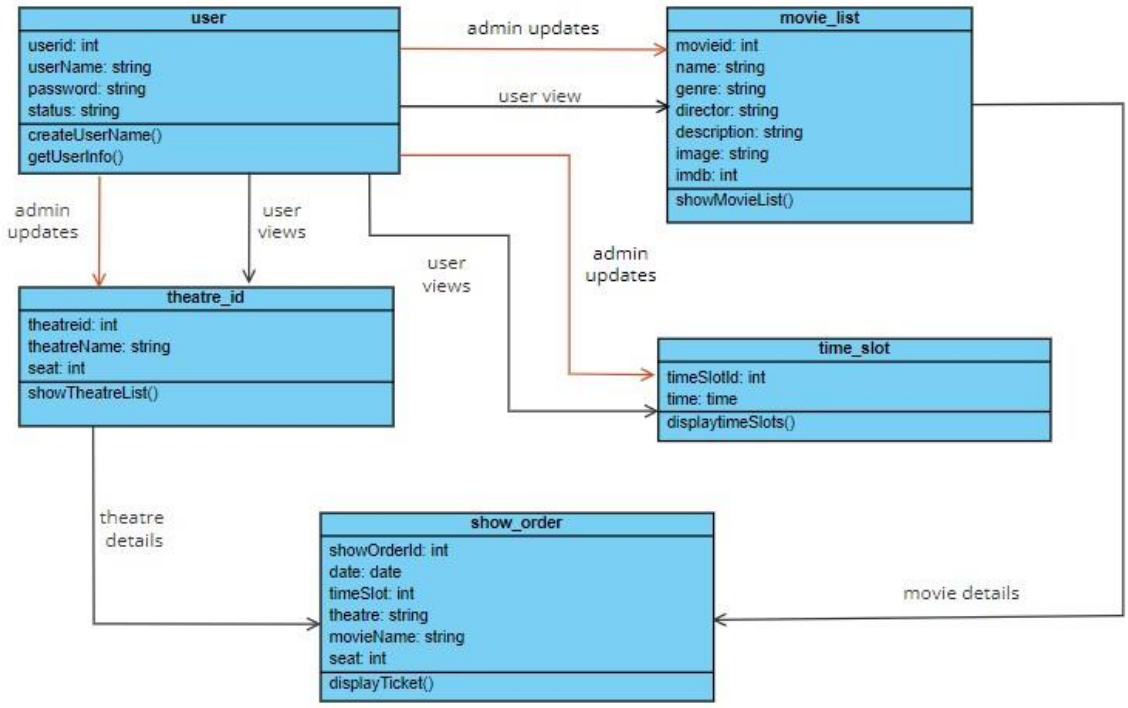


Figure: 4.3 Class Diagram

4.4 Entity-Relationship Diagram

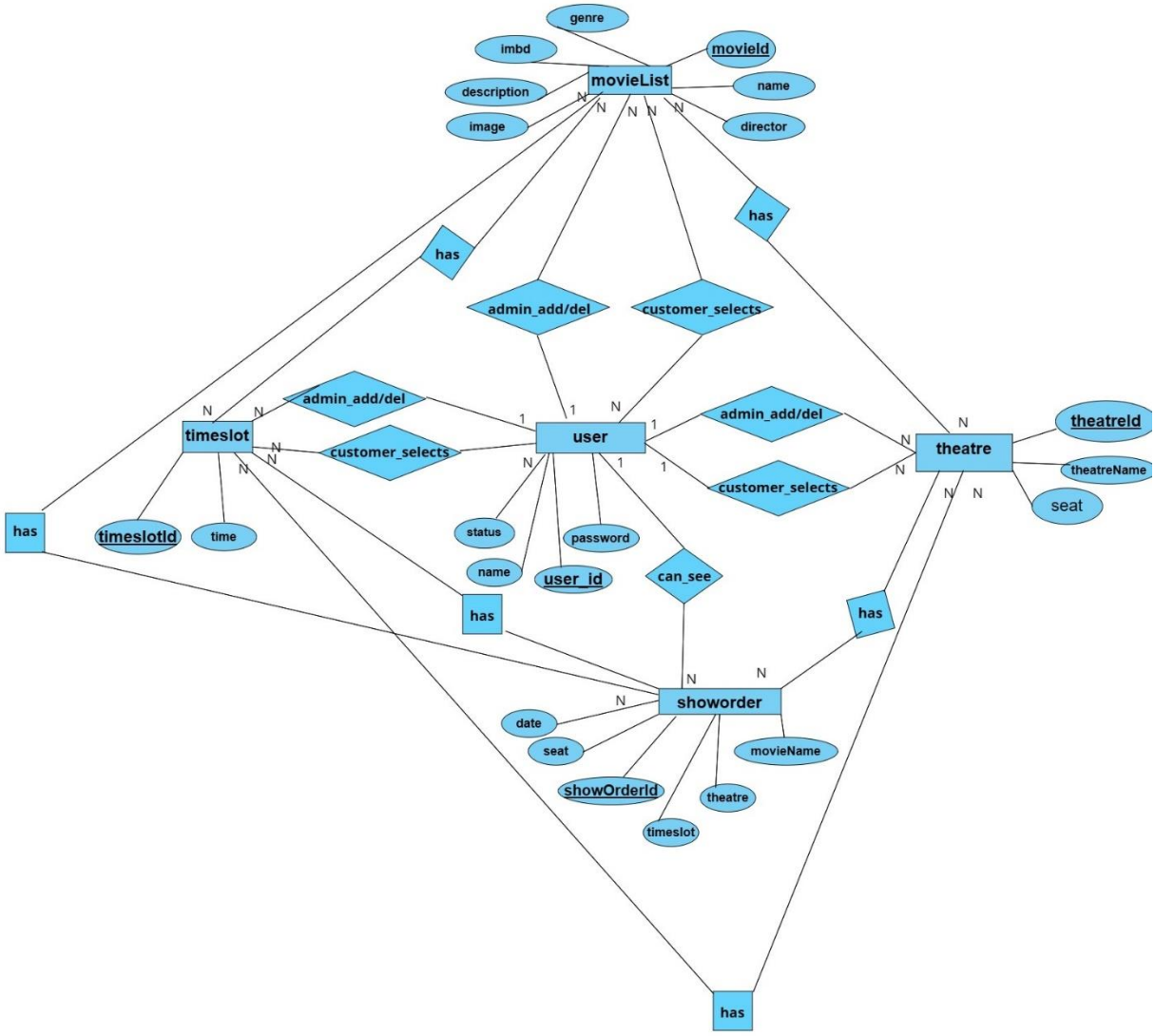


Figure 4.4: E-R Diagram of the Online Movie Booking System

4.5 Schema Diagram

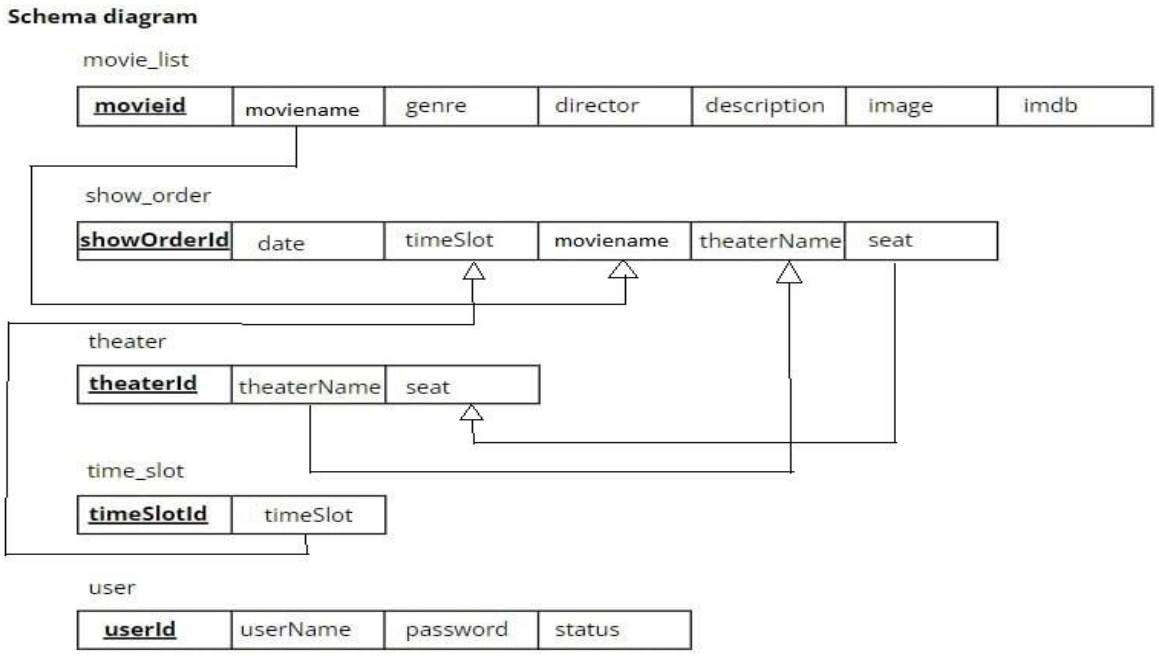


Figure 4.5: Schema Diagram of the Online Movie Booking System

CHAPTER 5

IMPLEMENTATION

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

Primary key- the field that is unique for all the record occurrences.

Foreign key-the field used to set relation between tables.

MySQL is multithreaded, multi user SQL database management System (DBMS). The basic program run as server providing multiuser access to a number of databases. MySQL is a database. The data in a MySQL is stored in a Database objects called tables. A table is a collection of related data entries and it consists of columns and rows. The databases are useful when storing information categorically.

5.1Creation of Tables

USER

```
CREATE TABLE `user` (
 `userId` int(11) NOT NULL,
 `userName` varchar(255) NOT NULL,
 `password` varchar(255) NOT NULL,
 `status` int(3) DEFAULT NULL
  );
```

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**userId**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**userName**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**password**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	**status**	int(3)			Yes	*NULL*			Change Drop More

Table 5.1: Creation of User Table

MovieList

```
CREATE TABLE `movielist` (
`movieId` int(11) NOT NULL,
`Name` varchar(255) DEFAULT NULL,
`Genre` varchar(25) DEFAULT NULL,
`Director` varchar(255) DEFAULT NULL,
`Description` varchar(255) DEFAULT NULL,
`image` varchar(255) DEFAULT NULL,
`imdb` varchar(255) DEFAULT NULL
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	movieId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	Name	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	Genre	varchar(25)	latin1_swedish_ci		Yes	NULL			Change Drop More
4	Director	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
5	Description	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
6	image	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
7	imdb	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More

Table 5.2: Creation of Movielist Table

Theatre

```
CREATE TABLE `theatre` (
`theaterId` int(11) NOT NULL,
`theaterName` varchar(255) DEFAULT NULL,
`seat` int(11) NOT NULL
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	theaterId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	theaterName	varchar(255)	latin1_swedish_ci		Yes	NULL			Change Drop More
3	seat	int(11)			No	None			Change Drop More

Table 5.3: Creation of Theatre Table

Timeslot

```
CREATE TABLE `timeslot` (
`timeslotId` int(11) NOT NULL,
`time` varchar(255) NOT NULL
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	timeslotId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	time	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Table 5.4: Creation of Timeslot Table

showorder

```
CREATE TABLE `showorder` (
  `showOrderId` int(11) NOT NULL,
  `date` varchar(255) NOT NULL,
  `timeslot` varchar(255) NOT NULL,
  `theatre` varchar(255) NOT NULL,
  `movieName` varchar(255) NOT NULL,
  `seat` varchar(255) NOT NULL
);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	showOrderId	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	date	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
3	timeslot	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
4	theater	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
5	movieName	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
6	seat	varchar(255)	latin1_swedish_ci		No	None			Change Drop More

Table 5.5: Creation of Showorder Table

5.2 Insertion Of Values:

Theatre

INSERT INTO `theatre` (`theaterId`, `theaterName`, `seat`) VALUES
(1, 'PVR Cineplex', 50),
(2, 'Cinepolis', 45),
(3, 'PVR Ansal Plaza', 60),
(4, 'Fun Cinemas', 70);

	theaterId	theaterName	seat
Edit Copy Delete	1	PVR Cinema	50
Edit Copy Delete	2	Cinepolis	45
Edit Copy Delete	3	Fun Cinemas	60
Edit Copy Delete	4	PVR Ansal Plaza	70

Table 5.6: Inserting values to the user Table

Timeslot

INSERT INTO `timeslot` (`timeslotId`, `time`) VALUES
(1, '11.00'),
(2, '2.00'),
(3, '5.00'),
(4, '8.00'),
(5, '9.00');

	timeslotId	time
Edit Copy Delete	1	11.00
Edit Copy Delete	2	2.00
Edit Copy Delete	3	5.00
Edit Copy Delete	4	8.00
Edit Copy Delete	5	9.00

Table 5.7: Inserting values in timeslot table

User

INSERT INTO `user` (`userId`, `userName`, `password`, `status`) VALUES
(1, 'admin', 'admin', 101),
(3, 'user', 'user', 202);

	userId	userName	password	status
Edit Copy Delete	1	admin	admin	101
Edit Copy Delete	3	user	user	202
Edit Copy Delete	6	test	test123	202
Edit Copy Delete	7	test1	121111	202

Table 5.8: Inserting values to the User Table

MovieList

INSERT INTO `movielist` (`movieId`, `Name`, `Genre`, `Director`, `Description`, `image`, `imdb`) VALUES
(1, 'Pathaan, 'Action', 'Christopher Nolan', 'description', '1.jpg', '6.5'),
(2, 'Doctor G', 'Adventure', 'Jon Watts', 'description', '2.jpg', '5.5'),

	movieId	Name	Genre	Director	Description	image	imdb
Edit Copy Delete	1	Brahmastra	Action	Christopher Nolan	Brahm?stra: Part One – Shiva (pronounced [b???ma?s...	batman.jpeg	5.5
Edit Copy Delete	2	Doctor G	Adventure	Jon Watts	Doctor G is a 2022 Indian Hindi-language comedy-dr...	2.jpeg	8.5
Edit Copy Delete	3	Pathaan	Action	David Fincher	Pathaan is a 2023 Indian Hindi-language action thr...	4.avif	8.8
Edit Copy Delete	4	Ram Setu	Action	Ridley Scott	Ram Setu (transl.?The bridge of Rama) is a 2022 In...	1.jpeg	7.2
Edit Copy Delete	5	Cuttputli	Adventure	Ranjit.M	Cuttputlli (transl.?Puppet) is a 2022 Indian Hindi...	6.jpeg	9
Edit Copy Delete	6	Raksha Bandhan	Comedy	Anand.L	Raksha Bandhan (transl.?The bond of protection) is...	7.jpeg	6.4

Table 5.9: Inserting values to Movielist Table

Showorder

INSERT INTO `showorder` (`showOrderId`, `date`, `timeslot`, `theatre`, `movieName`, `seat`) VALUES
(1, '2023-01-30', '11.00', PVR', Pathaan', '49'),
(2, '2023-01-31', '5.00', 'BlockBluster', 'Gladiator', '49'),
(3, '2023-01-01', '2.00', 'Balaka Cineplex', 'Spider-Man: Homecoming (2017)', '48'),
(4, '2023-01-26', '8.00', 'Balaka Cineplex', 'Batman Begins', '49'),
(5, '2023-01-23', '11.00', 'Basundhara Cineplex', 'Batman Begins', '50'),
(6, '2023-01-27', '2.00', 'Basundhara Cineplex', 'Batman Begins', '48'),
(7, '2023-1-26', '11.00', 'Shamoly Cineplex', 'Batman Begins', '50');

	showOrderId	date	timeslot	theater	movieName	seat
Edit Copy Delete	1	2023-01-30	11.00	Basundhara Cineplex	Batman Begins	49
Edit Copy Delete	2	2023-01-31	5.00	BlockBluster	Gladiator	49
Edit Copy Delete	3	2023-01-24	2.00	Balaka Cineplex	Spider-Man: Homecoming (2017)	48
Edit Copy Delete	4	2023-01-26	8.00	Balaka Cineplex	Batman Begins	49
Edit Copy Delete	6	2023-01-27	2.00	Basundhara Cineplex	Batman Begins	48
Edit Copy Delete	9	2023-01-30	11.00	PVR Cinema	Brahmastra	49
Edit Copy Delete	10		11.00	PVR Cinema	Doctor G	49

Table 5.10: Inserting values to showorder Table

CHAPTER 6:

SOFTWARE TESTING

Table 6.1:Test Cases

Test no.	**Test** Name	**Expected output**	**Actual output**	**Status**
1	Login	Successfully logged in	Successfully logged in	Pass
2	Searching for movie	Movie details must be shown	Movie details is shown	Pass
3	Customer Booking Ticket	Ticket should be booked if available	Ticket is booked	Pass
4	Admin adding new movie	New Movie should be added	New movie is added	Pass
5	Admin adding new Theatre	New theatre should be added	New theatre is added	Pass
6	Admin adding time slots	New Time slot should be added	New Time Slot is added	Pass

CHAPTER 7:

CONCLUSION

Online Movie Booking System is a PHP/MYSQL-based web application. Its main goal is to help users book movie tickets online without going to nearby cinemas. Also, the users have several options in which they can select a suitable time and theatre to book movie tickets for their favourite movies they want to watch.

FUTURE SCOPE

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion. The following are the future scope for the project.

1. Change in UI design
2. Adding More Theatre & Movies
3. Adding Options to Book Event, Plays, Sports & More
4. Easy to Use

APPENDICES

A. SAMPLE CODE

Database Connection:

```php
<?php

$host="localhost";
$username="root";
$password="";
$db_name="movieDb";
// Create connection
$conn = new mysqli($host, $username, $password,$db_name);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
} else{

}
?>
```

Login:

```php
<?php
if (!session_id()) {
  session_start();
}
include_once ('db.php');


$user=$_REQUEST['postName'];
$pass=$_REQUEST['postPassword'];



$sql="select userId,status from user where  userName ='$user' and password='$pass';";

$account_type="";

$res=$conn->query($sql);
if (($data=$res->fetch_object())) {
  $_SESSION['user']= $data->userId ;
  $account_type=$data->status;


  if (  $account_type ==="101") {
    echo "1";
  }else if ($account_type ==="202"){
    echo "2";
  }
}else{
  echo "3";
}


?>
```

Admin adding Theatre:

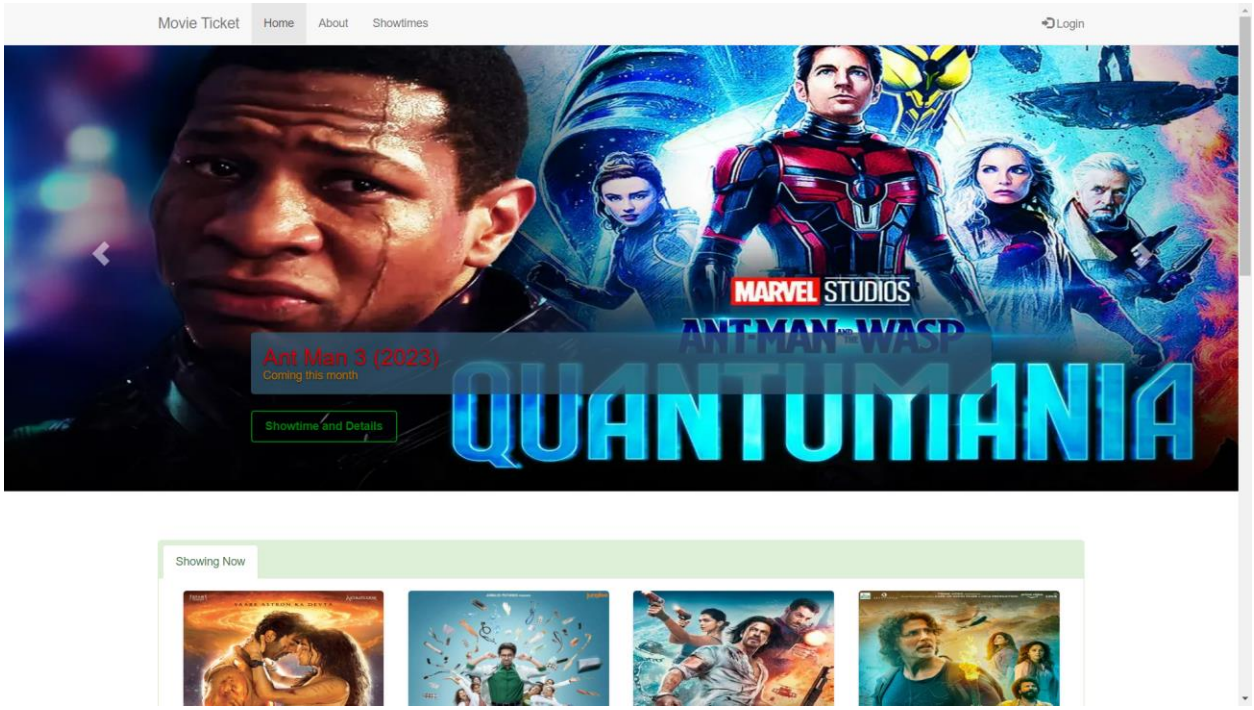
```php
<?php
if (isset($_POST['submit'] )&& !empty($_POST['submit']))

{
    $TheaterName=$_POST['TheaterName'];
    if (!(empty($TheaterName) ))
    {

        $var=new AddProduct();
        $var->productAdd($conn);
    }
}
else{

}
?>

<?php
class AddProduct{
    public function productAdd($conn)
    {
        $sql="insert into theater(theaterId,theaterName) values('',?);";

        if(($stmt=$conn->prepare($sql))) {
            $stmt->bind_param("s",$theaterName);

        }else
        {
            var_dump($conn->error);
        }
        $theaterName=$_POST['TheaterName'];
        $stmt->execute();
        $stmt->close();
        $_SESSION['msg']="Theater Successfully  Added";
        header ("Location: adminpage.php" );
    }
}
?>
```

Admin adding Timeslot:

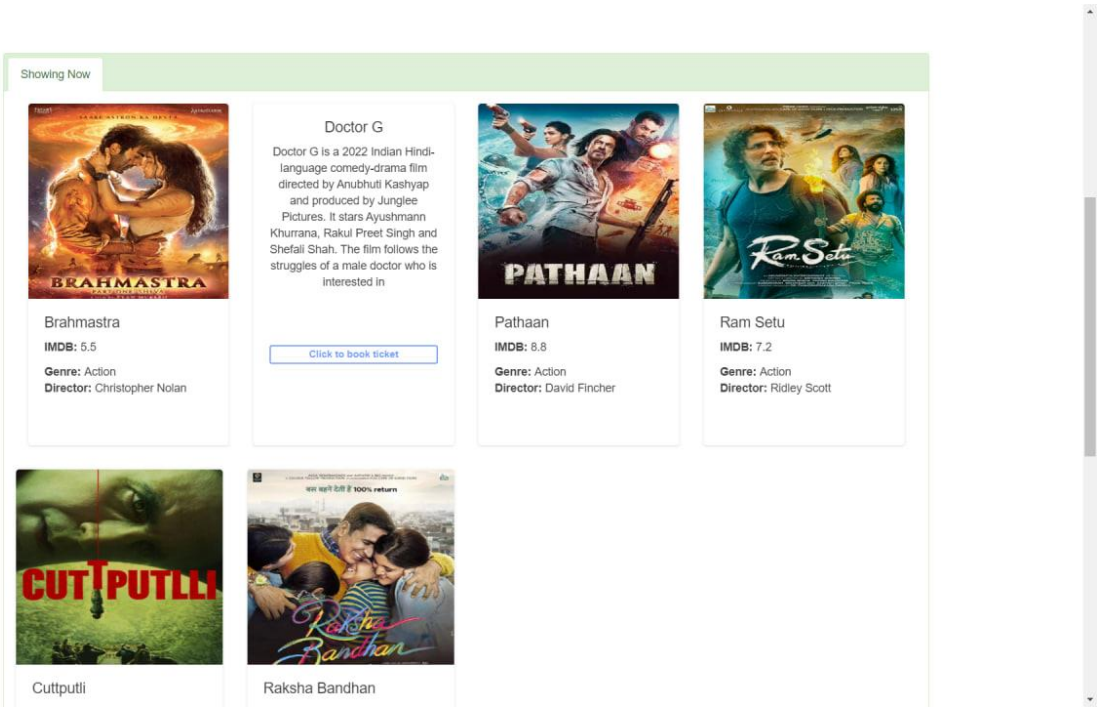
```php
<?php
if (isset($_POST['submit'] )&& !empty($_POST['submit']))

{
    $timeSlot=$_POST['timeSlot'];
    if (!(empty($timeSlot) ))
    {
        $var=new AddProduct();
        $var->productAdd($conn);
    }
}

?>

<?php
class AddProduct{
    public function productAdd($conn)
    {
        $sql="insert into timeSlot(timeslotId,time) values('',?);";

        if(($stmt=$conn->prepare($sql))) {
            $stmt->bind_param("s",$timeSlot);

        }else
        {
            var_dump($conn->error);
        }
        $timeSlot=$_POST['timeSlot'];
        $stmt->execute();
        $stmt->close();
        $_SESSION['msg']="Time Slot Successfully Added";
        header ("Location: adminpage.php" );
    }
}
?>
```

B. SNAPSHOTS

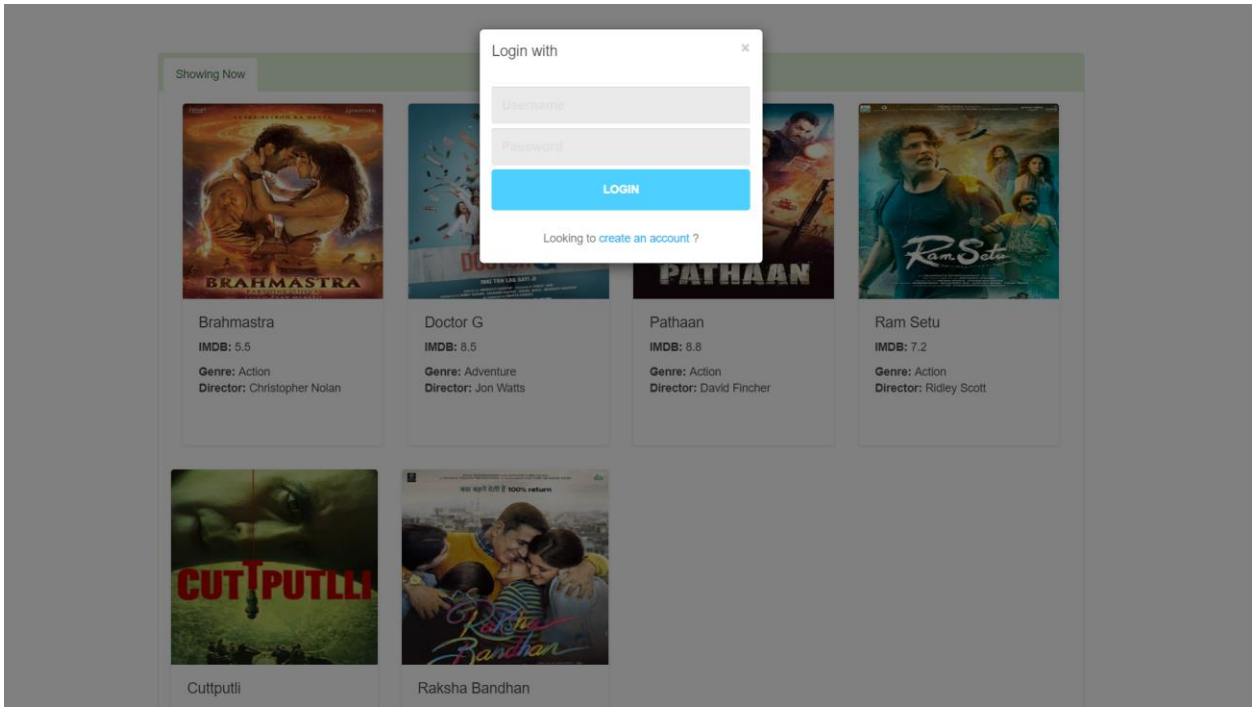
Home Page:



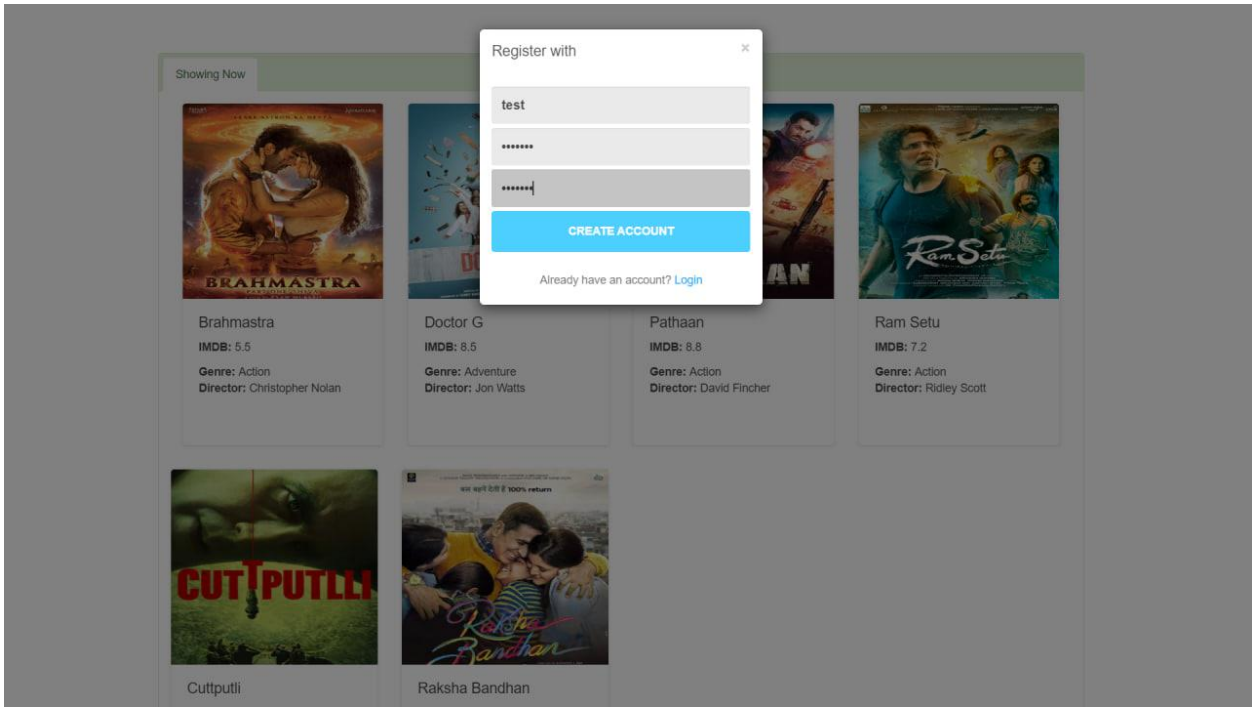
Movies List:



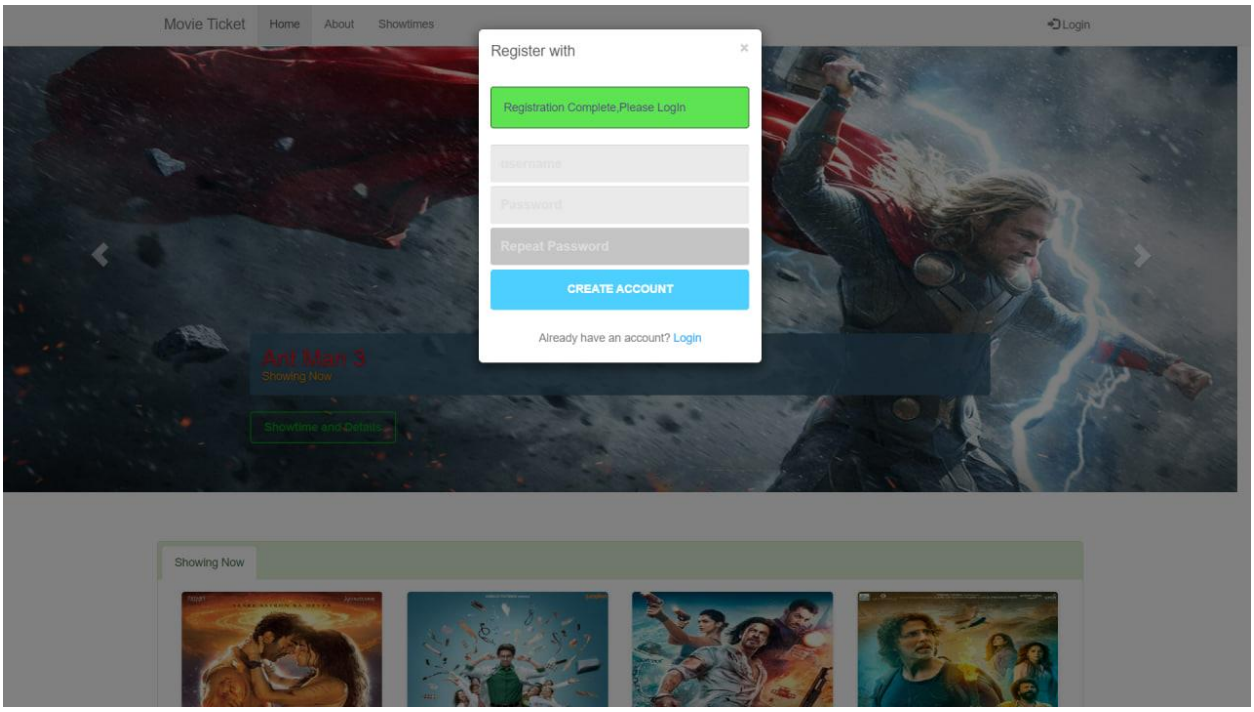
Login Panel:



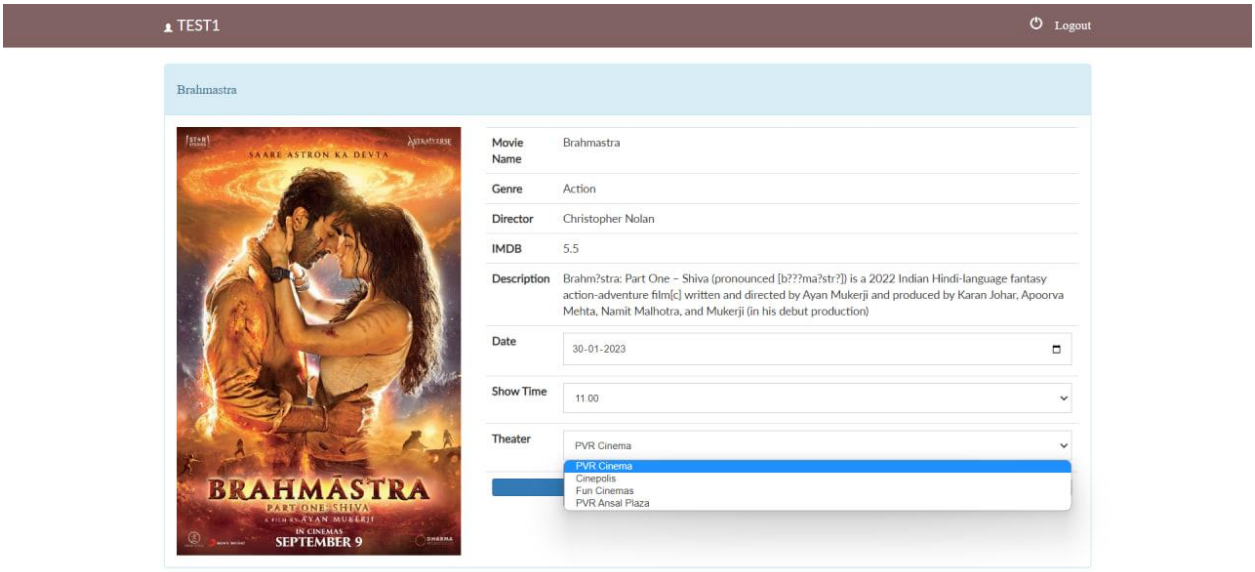
Registration Panel:



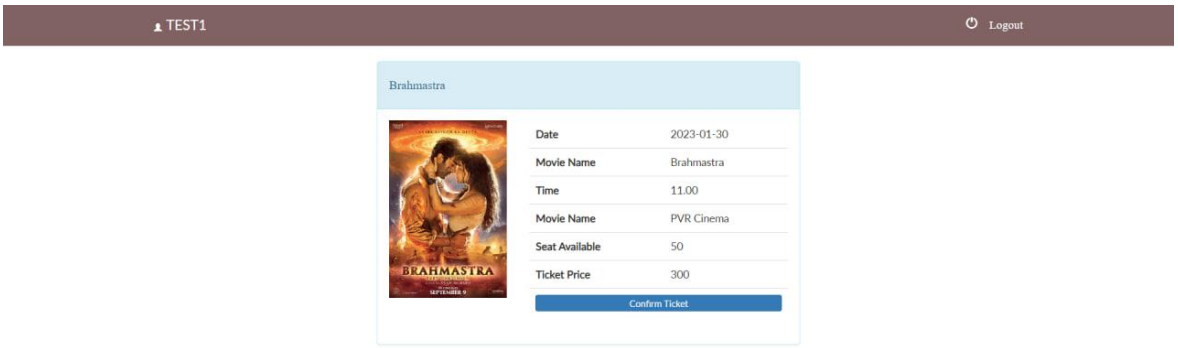
User Registered Successfully:



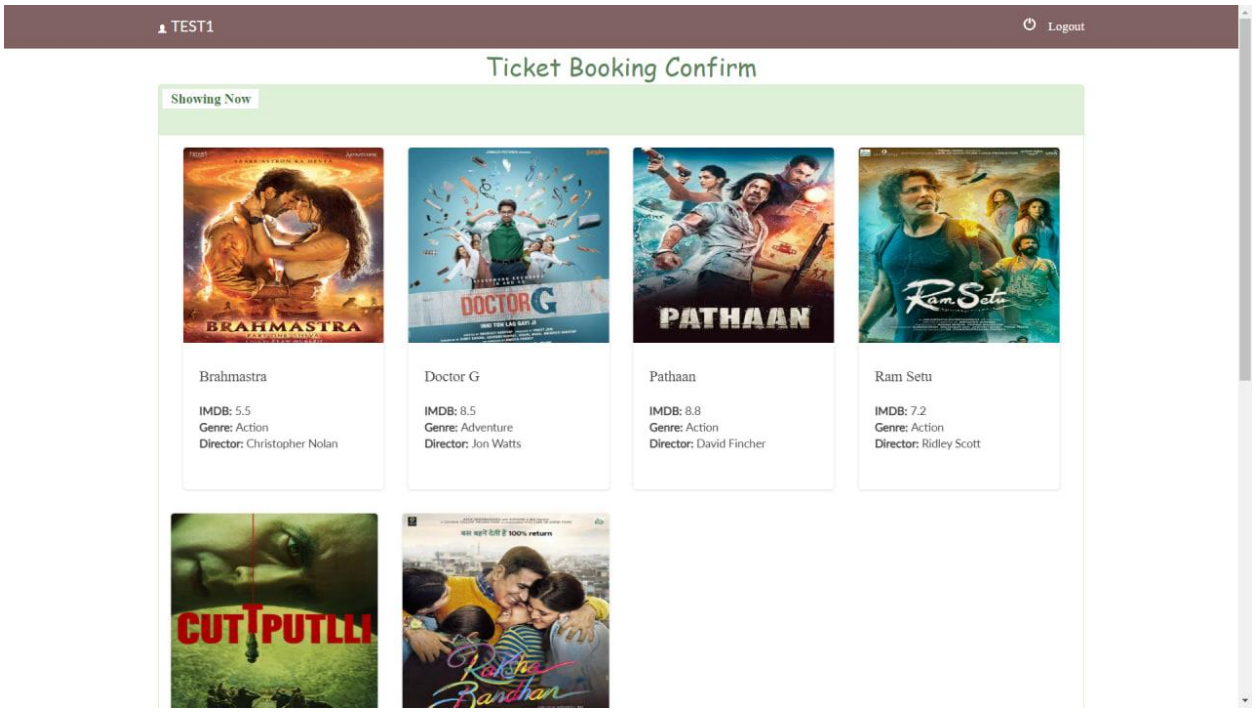
Booking Movie Ticket::



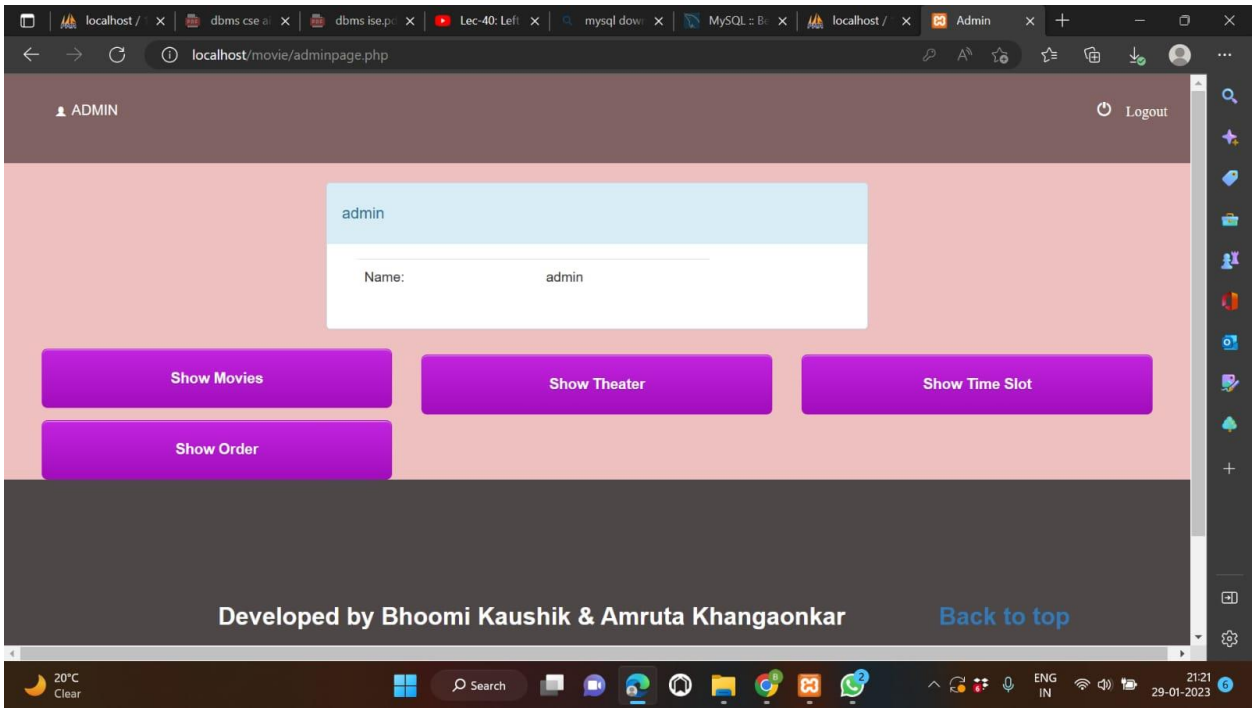
Confirming Movie Ticket:



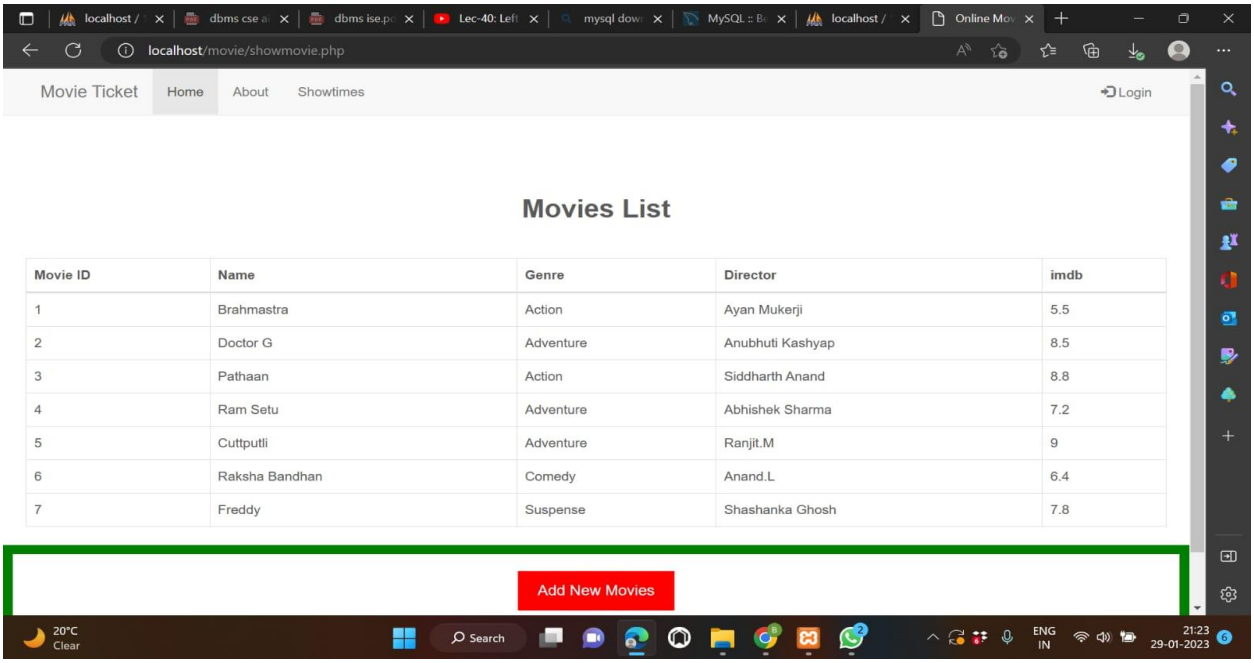
Movie Ticket Confirm:



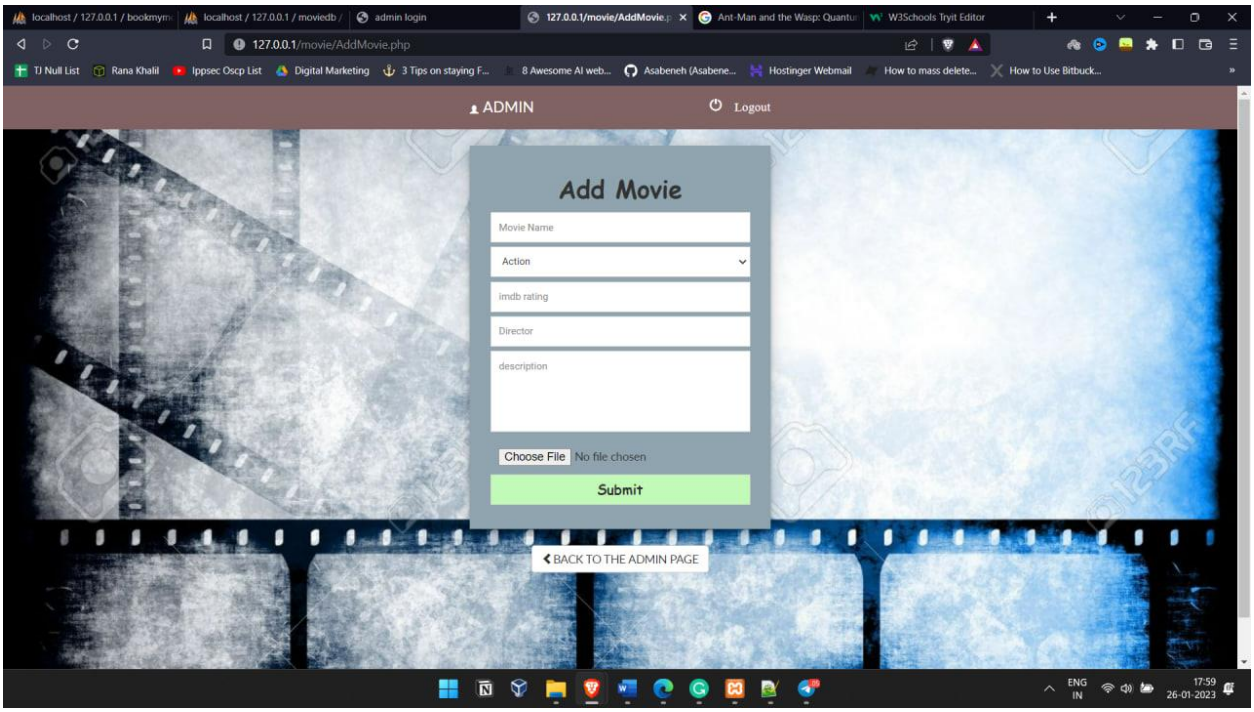
Admin Panel:



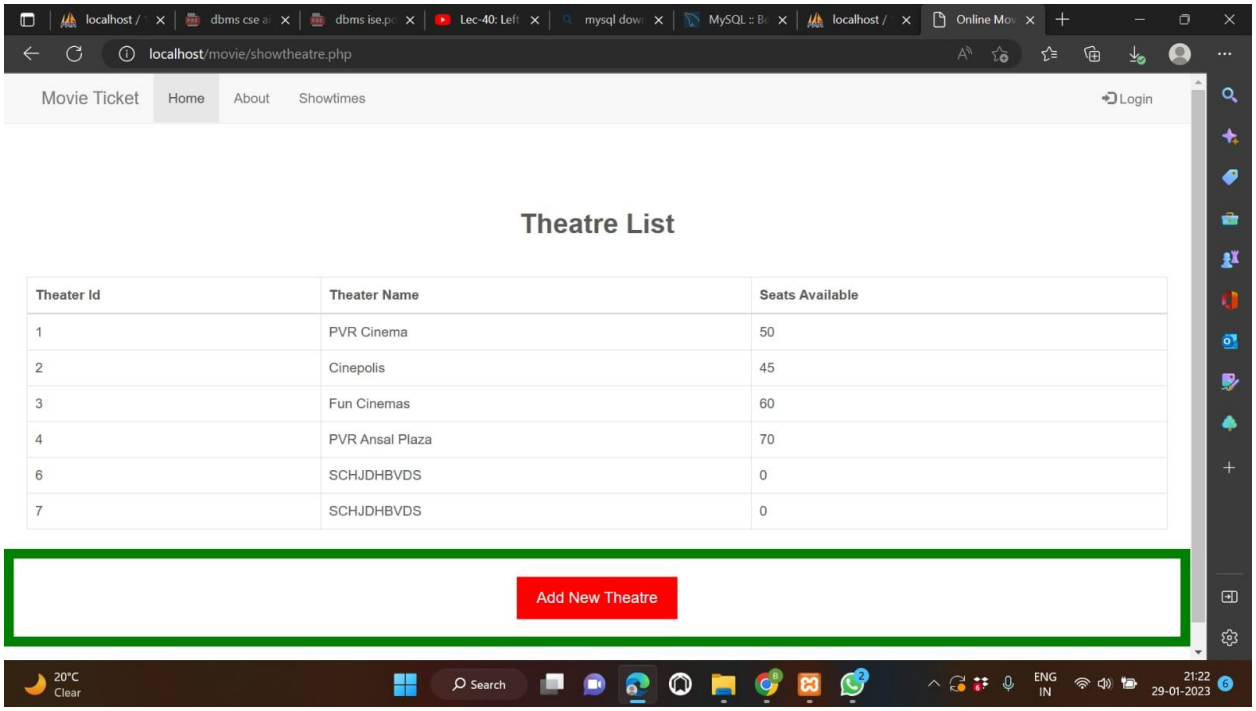
Movie List:



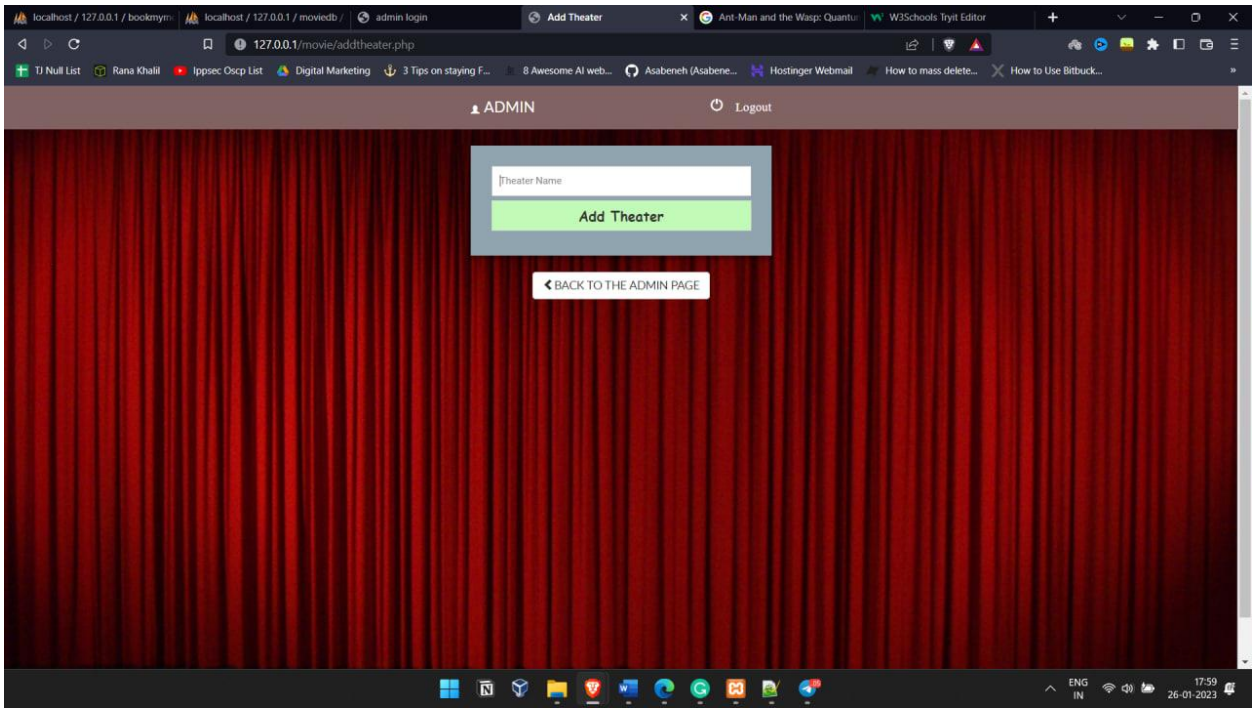
Add Movie:



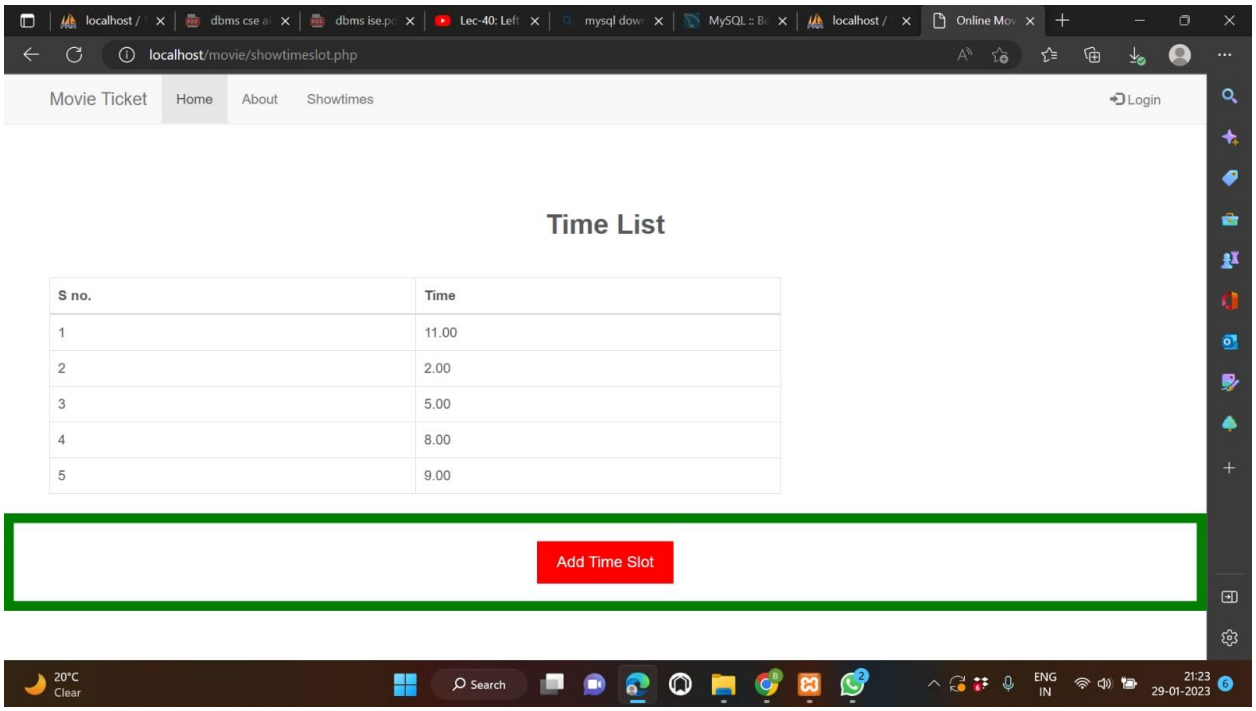
Theatre List:



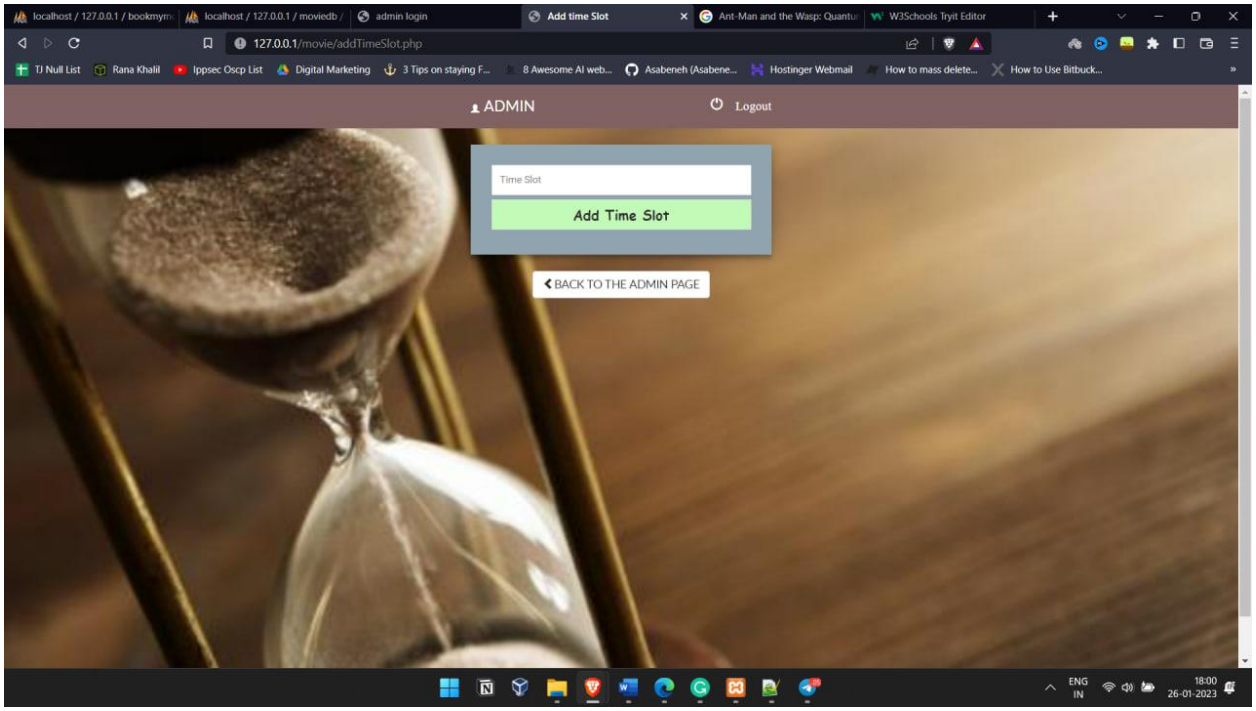
Adding Theatre as Admin:



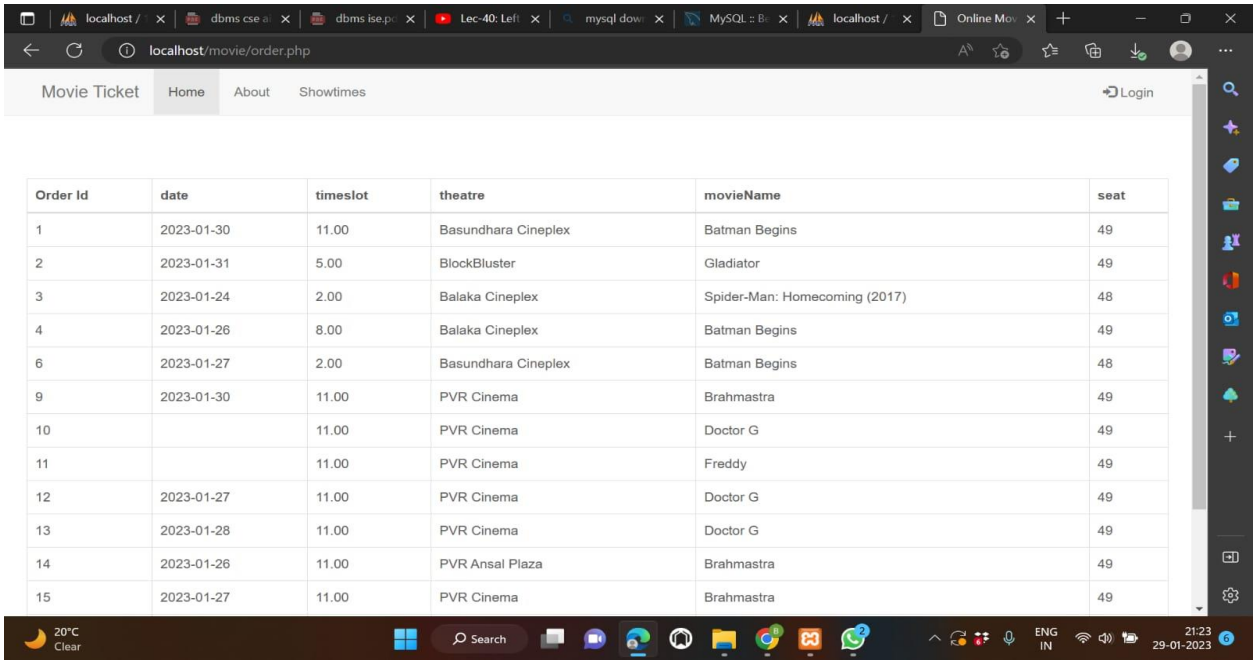
Time Slot List:



Adding Time Slot as Admin:

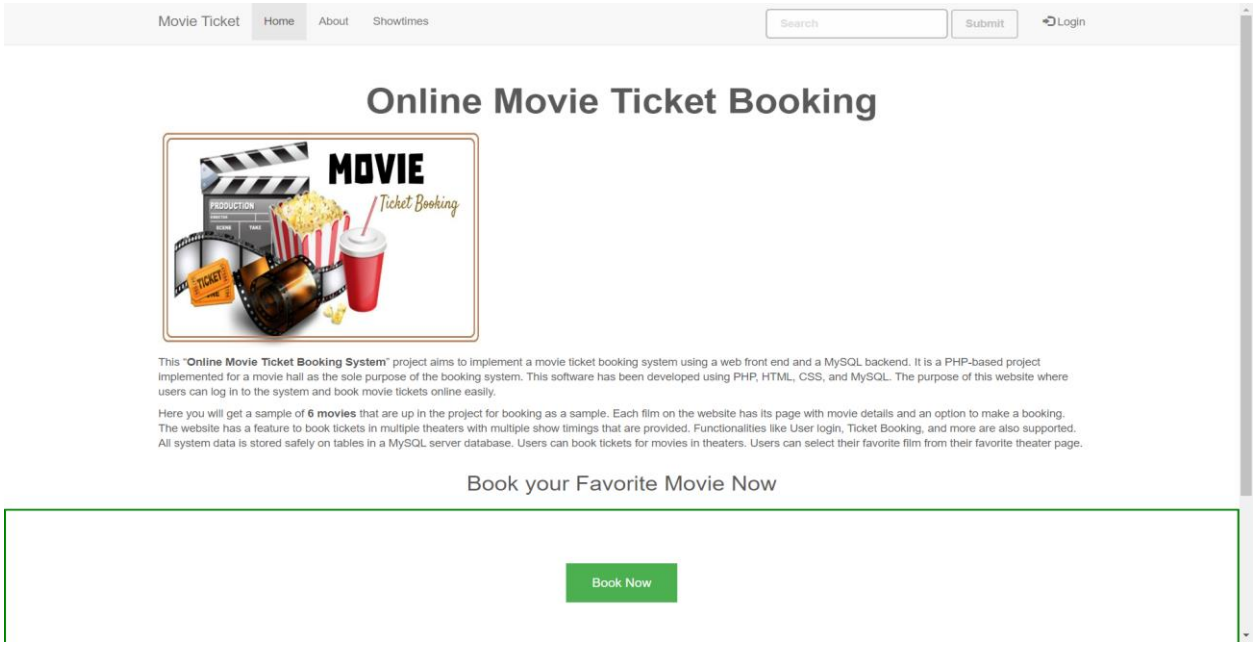


Order List:



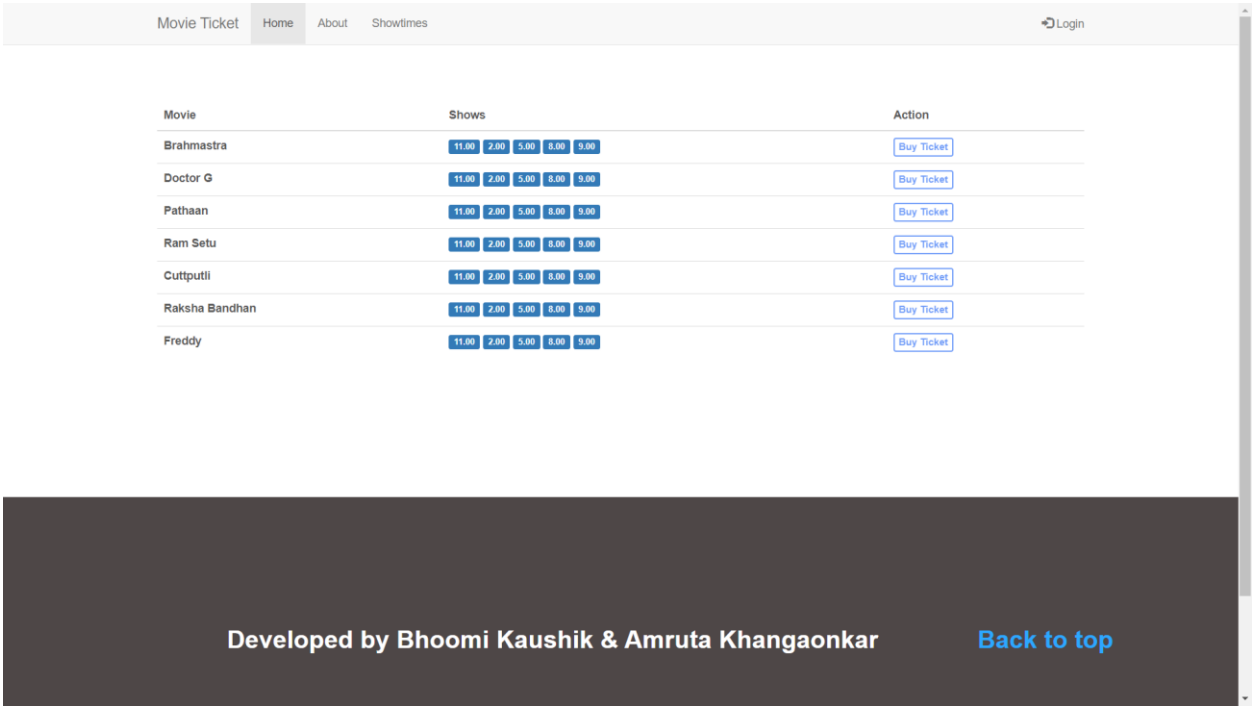
Order Id	date	timeslot	theatre	movieName	seat
1	2023-01-30	11.00	Basundhara Cineplex	Batman Begins	49
2	2023-01-31	5.00	BlockBluster	Gladiator	49
3	2023-01-24	2.00	Balaka Cineplex	Spider-Man: Homecoming (2017)	48
4	2023-01-26	8.00	Balaka Cineplex	Batman Begins	49
6	2023-01-27	2.00	Basundhara Cineplex	Batman Begins	48
9	2023-01-30	11.00	PVR Cinema	Brahmastra	49
10		11.00	PVR Cinema	Doctor G	49
11		11.00	PVR Cinema	Freddy	49
12	2023-01-27	11.00	PVR Cinema	Doctor G	49
13	2023-01-28	11.00	PVR Cinema	Doctor G	49
14	2023-01-26	11.00	PVR Ansal Plaza	Brahmastra	49
15	2023-01-27	11.00	PVR Cinema	Brahmastra	49

AboutUs:



About Us

ShowTiming Page:



BIBLIOGRAPHY

Web references

[1] Fundamentals Of Database Systems by Ramez Elmarsi and Shamkant B. Navathe 7th edition

[2] DBMS: Ramakrishnan - Database Management Systems 3rd Edition

[3] PHP & MySQL Web development by Luke Welling & Laura Thompson

[4] www.php.net

[5] www.mysql.com

[6] www.xamppserver.com

[7] www.w3schools.com

[8] www.stackoverflow.com