**ONLINE MOVIE TICKET BOOKING SYSTEM**

**PROJECT REVIEW** - **I**

**DATE:**

**Submitted by:**

**NAME:** G SANKARA SUBRAMANIAN

**REG NO:** 23PCA507

**CLASS:** MCA II - YEAR

**Internal Project Guide:**

## Dr. A. REGITA THANGAM MCA,M.PHIL,PH.D,SET

## ACKNOWLEDGEMENT

At first I extend my deepest gratitude to the **ALMIGHTY** for providing me all the requirements for completing this project.

I extend my deepest sense of gratitude to **Rev. Fr. Dr. Godwin Rufus S.J.,** Principal, St. Xavier’s College (Autonomous), Palayamkottai, **Dr. A. Lordusamy**, Deputy Principal (Shift-II) and **Dr. S. Chidambaranathan** and **Mrs. A. Dhana Praveena**, Vice Principals (Shift-II) for having permitted me to carry out this project work.

No one grows tired to thank, **Dr. S. Saraswathi**, M.C.A., M.Phil., Ph.D., Head of the Department, Department of Computer Applications, who has been a source of inspiration, all time motivator and played a major role in the conduct of my project work.

I wish to express my deep sense of gratitude to my Internal Guide**, Dr. A. Regita Thangam**, M.C.A., M.Phil., Ph.D., SET., for the guidance and useful suggestions, which helped me in completing the project work on time.

Words are inadequate in offering my thanks to all the **faculty members** of the Department of Computer Applications, for their encouragement and cooperation in carrying out the project work.

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved

**Parents** for their blessings and wishes for the successful completion of this project.

**23PCA507**

**G SANKARA SUBRAMANIAN**

**ABSTRACT**

This Project proposes the Development of an Online Movie Ticket Booking System that offers a convenient and user-friendly platform for customers to reserve movie tickets. The Online Movie Ticket Booking System is a Web-Based Application designed to simplify the task of booking movie tickets online. The system will eliminate the need to physically visit theaters to purchase tickets, saving users time and effort. It will also provide Theatre Management with an efficient tool to manage showtimes, bookings and customer data. In conclusion, the Online Movie Ticket Booking System enhances convenience for moviegoers by providing a user-friendly platform for booking tickets online, thereby contributing to the efficiency and profitability of cinema operations

**Backend and Frontend Development Technologies:**

The system developed for online using PHP, Html, Css, Javascript as front end, and Bootstrap as Framework, and MySQL as back end.

# TABLE OF CONTENTS

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SL.NO.** | **DESCRIPTION** | **PAGE NO.** |
| 1 | INTRODUCTION | 1 |
| 2 | SYSTEM STUDY   * 1. Existing System   2. Proposed System   3. Problem Definition and Project Description | 2  2 |
| 3 |
| 3 | SYSTEM ANALYSIS   * 1. Requirements Specification   2. Feasibility Study | 4  5 |
| 4 | SYSTEM DESIGN   * 1. Architectural Design   2. Data Flow Diagram   3. Data Dictionary   4. User Interface Design   5. Normalization | 7 |
| 8  10 |
| 14 |
| 17 |
| 5 | SYSTEM TESTING   * 1. Types of Testing   2. Types of Validations   3. Error Messages | 19  21  22 |
| 6 | USER MANUAL   * 1. Installation Manual   2. Operational Manual | 24  24 |
| 7 | SYSTEM IMPLEMENTATION  7.1 Special features of the languages | 26 |
| 8 | FUTURE ENHANCEMENT | 30 |
| 9 | CONCLUSION | 32 |
| 10 | BIBLIOGRAPHY | 34 |
| 11 | APPENDIX   * 1. Sample Screen Layouts   2. Sample Codings | 36  44 |

# INTRODUCTION

ONLINE MOVIE TICKET BOOKING SYSTEM

## INTRODUCTION

An **Online Movie Ticket Booking System** in PHP is a web-based application designed to allow users to browse available movies, select showtimes, book tickets, and make payments online. New users can sign up by providing personal details like name, email, phone number, and password. The system simplifies the traditional process of purchasing movie tickets by providing a user-friendly interface where customers can view movie details and complete transactions from the comfort of their homes. Users can create an account to manage their bookings and receive personalized recommendations. Secure login mechanisms protect user data.

Displays a list of currently available movies, complete with details such as synopsis, cast and duration Search options to help users find movies by title. Admins can manage the scheduling of movies by setting showtimes, assigning them to specific screens, and managing their availability. Users can view available showtimes for each movie and select their preferred time slot. Admin panel allows cinema staff to manage movie schedules, seat availability, and user queries. Admins can add, edit, or delete movie details, manage showtimes, and track bookings Users can book tickets anytime, anywhere, avoiding long queues at the cinema.

A clean, intuitive interface is crucial for enhancing user experience. Key actions, like browsing movies or booking tickets, should be easy to access. Streamlines the ticketing process, reducing manual errors and improving overall customer satisfaction. This project is an excellent opportunity to apply your PHP and web development skills to create a practical, real-world application that addresses a common consumer need.

Department of Computer Applications, St. Xavier’s College (Autonomous), Palayamkottai Page 1

# SYSTEM STUDY

## SYSTEM STUDY

* 1. **EXISTING SYSTEM**

Present system is manual. Customers have to visit the cinema physically or make phone calls to

book tickets, which is inconvenient and time-consuming.

The following are the disadvantages of the existing system:

* + - Manual systems cannot provide real-time updates on showtimes.
    - Searching for and retrieving specific booking records or customer information from manual

registers can be slow and inefficient.

* + - Maintaining records of booked tickets, customer details, and payment information manually

in registers or spreadsheets is cumbersome and prone to errors.

* + - Manual records often contain redundant data and are more susceptible to human errors
    - Manual records tend to contain data which are redundant.

## PROPOSED SYSTEM

The existing manual system is replaced with a web application changing the way of

maintaining customers which proves to be beneficial.

The following are the advantages of the proposed system:

* + - Customers can book tickets anytime, anywhere, without the need to physically visit the cinema.
    - The user-friendly interface and easy navigation make it simple for customers to browse movies.
    - All possible features such as verification, validation, security, etc., have been considered.
    - The online system is available 24/7, allowing customers to book tickets at any time
    - The system automatically records all bookings, customer details, and payment information in a

centralized database.

## PROBLEM DEFINITION AND PROJECT DESCRIPTION

The project titled as “ONLINE MOVIE TICKET BOOKING SYSTEM” is a web-based application that provides a convenient platform for customers to book movie tickets online. The manual process of booking tickets at physical counters is time-consuming and inefficient, leading to long queues, errors, and customer dissatisfaction. To streamline this process and enhance the customer experience, the “ONLINE MOVIE TICKET BOOKING SYSTEM” will be of great benefit. This software has different modules that ensure the smooth functioning of the system, and they are

* Admin
* Theatre Assistance
* User

#### ADMIN MODULE

* The Admin can view the details of all movies currently listed on the platform.
* The Admin can remove movies.
* Add Theatre: In this section, admin can add theatre.
* Upcoming Movie News:The administrator can add upcoming movie news

#### THEATRE ASSISTANCE MODULE

* Theatre Assistance can add and view movies and shows.
* Theatre Assistance can view total bookings and shows.
* Theatre Assistance can manage Theatre Details (Add Screen/Show Times).

#### USER MODULE

* Users can create an account by providing basic information such as name, email, phone number

and password.

* Registered users can log in to their accounts using their credentials.
* Users can view Upcoming Movie News and Movie trailers.
* Users can browse through the list of currently available movies.

# SYSTEM ANALYSIS

## SYSTEM ANALYSIS

#### REQUIREMENTS SPECIFICATION HARDWARE REQUIREMENTS

Processor : AMD

Ram : 4 GB (3.89 GB usable)

Hard disk : 40 GB

Monitor : 15’’mVGA monitor

Keyboard : 104 keys Keyboard

Mouse : Optical mouse

#### SOFTWARE REQUIREMENTS

Operating System : 64-bit OS

Local Host Server : Xampp Server Database Connectivity : PhpMyAdmin Front End : Html and PHP

Back End : MYSQL

## FEASIBILITY STUDY

A feasibility analysis is conducted to decide if the solution considered to meet the criteria is feasible and workable in the software. During the feasibility study, information such as resource availability, cost estimates for software production, advantages of the software to the enterprise after its development, and cost to be expended on its maintenance is determined. The feasibility study aims to ascertain why developing software is appealing to users, adaptable to change, and compliant with applicable requirements. The system has been tested for feasibility in the following points:

* + - Technical Feasibility
    - Operational Feasibility
    - Economic Feasibility

#### TECHNICAL FEASIBILITY

Technical feasibility evaluates the available infrastructure (such as hardware and software) and technologies needed to meet the user needs of software under time and budget constraints. The following are the activities often performed by technical feasibility.

* Examines whether there are technical guarantees of accuracy, reliability, ease of access and data security.
* Determines whether the application infrastructure is well-established.
* Ensures whether the proposed system provides adequate response to inquiries, regardless of the number or location of users

The current system developed is technically feasible as it provides the technical guarantee of accuracy, reliability, security and easy access to the users.

#### OPERATIONAL FEASIBILITY

The proposed system is beneficial only if it can be turned out into information system which will meet the operating requirements of the organization. The extent to which the required software completes a sequence of steps to address challenges and requirements of the developer and users respectively is measured by operational viability. The following are the operations carried out by operational feasibility:

* + Determines whether sufficient support for the organization is provided from the users.
  + Ensures proper working of the system if it is being developed and implemented.
  + Checks whether there will be any resistance from the users that will ruin the possible benefits of the application

This Online Movie Ticket Booking System would ensure the optimal utilization of computer resources and would help in the improvement of performance status.

#### ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. Economic feasibility needs to consider the expenses made on purchasing, such as hardware purchasing and required activities to carry out software development. It is also necessary to consider the benefits that can be achieved by developing the software. Software is economically feasible when it focuses on the issues listed below.

* + - Expense incurred on software development for achieving long-term gains for an organization.
    - Expenses required to conduct elicitation and requirements analysis
    - Hardware and software cost, development team, and training cost.

This system is economically feasible. Since this system is developed using the existing resources and technologies, there is nominal expenditure which ensures the economic feasibility of the system.

# SYSTEM DESIGN

* 1. **ARCHITECTURAL DESIGN**

1. **SYSTEM DESIGN**

Book a Ticket

View Showing Movies

User Login

View Movie Trailers

User Registration

View Upcoming Movies

Theatre Details

Add Show

View Movies

Add Movie

Add Upcoming Movie News

View Today Bookings

View Available Shows

View/Manage Shows

View Running Movies

Add Theatre

View Movies List

**USER**

**THEATRE**

**ASSISTANCE**

**ADMIN**

**ONLINE MOVIE TICKET BOOKING SYSTEM**

#### DATA FLOW DIAGRAM DFD Level 0

Theatre Assistance

Online Movie Ticket Booking System

Database

User

Admin

**DFD Level 1**

tblnews

Add Upcoming Movie News

tbltheatre

Add Theatre

tblmovie

View Movies List

Admin

tblmovie

Add/View Movies

View Showing Movies

tblmovie

tbllogin

tblbookings

Book a Ticket

tblregistration

User Registration

User Login

User

Theatre Assistance

tblscreens

Theatre Details

View Today Bookings

tblshowtime

View Available Shows

tblshows

Add/View/Manage Shows

#### DATA DICTIONARY

**Table Name:** tblbookings

**Purpose:** To Book a Movie Ticket

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | book\_id | int | 11 | Primary Key | Booking Id |
| 2 | ticket\_id | varchar | 30 | Not Null | Ticket Id |
| 3 | t\_id | int | 11 | Not Null | Theatre Id |
| 4 | user\_id | int | 11 | Not Null | User Id |
| 5 | show\_id | int | 11 | Not Null | Show Id |
| 6 | screen\_id | int | 11 | Not Null | Screen Id |
| 7 | no\_seats | int | 3 | Not Null | Number of Seats |
| 8 | amount | int | 5 | Not Null | Amount |
| 9 | ticket\_date | date |  | Not Null | Ticket Date |
| 10 | date | date |  | Not Null | Date |
| 11 | status | int | 1 | Not Null | Confirmation |

**Table Name:** tbllogin

**Purpose:** To store the login information of users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | id | int | 11 | Primary Key | Id |
| 2 | user\_id | int | 11 | Not Null | UserId |
| 3 | username | varchar | 50 | Not Null | UserName |
| 4 | password | varchar | 50 | Not Null | Password |
| 5 | user\_type | int | 1 | Not Null | Type of Users |

**Table Name:** tblmovie

**Purpose:** To store the details of Movies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | movie\_id | int | 11 | Primary Key | Movie Id |
| 2 | t\_id | int | 11 | Not Null | Theatre Id |
| 3 | movie\_name | varchar | 30 | Not Null | Name of the Movie |
| 4 | cast | varchar | 10 | Not Null | Cast of Movie |
| 5 | desc | varchar | 20 | Not Null | Description of Movie |
| 6 | release\_date | date |  | Not Null | Release Date |
| 7 | image | varchar | 50 | Not Null | Movie Image |
| 8 | video\_url | varchar | 50 | Not Null | Movie Trailer |
| 9 | status | int | 1 | Not Null | Confirmation |

**Table Name:** tblnews

**Purpose:** To store the details of Upcoming Movie News

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | news\_id | int | 11 | Primary Key | News Id |
| 2 | name | varchar | 30 | Not Null | Name of the Movie |
| 3 | cast | varchar | 30 | Not Null | Cast |
| 4 | news\_date | date |  | Not Null | Release Date |
| 5 | description | varchar | 40 | Not Null | Description of Movie |
| 6 | attachment | varchar | 50 | Not Null | Movie Image |

**Table Name:** tblregistration

**Purpose:** To store the details of the Registered Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | user\_id | int | 11 | Primary Key | User Id |
| 2 | name | varchar | 50 | Not Null | Name |
| 3 | email | varchar | 50 | Not Null | Email |
| 4 | phone | varchar | 12 | Not Null | Phone Number |
| 5 | age | int | 2 | Not Null | Age |
| 6 | gender | varchar | 10 | Not Null | Gender |

**Table Name:** tblscreen

**Purpose:** To store the details of the Screens

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | screen\_id | int | 11 | Primary Key | Screen Id |
| 2 | t\_id | int | 11 | Not Null | Theatre Id |
| 3 | screen\_name | varchar | 50 | Not Null | Screen Name |
| 4 | seats | int | 11 | Not Null | No of Seats |
| 5 | charges | int | 11 | Not Null | Ticket Price |

**Table Name:** tblshowtime

**Purpose:** To store the details of the Showtime

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | st\_id | int | 11 | Primary Key | Show Time |
| 2 | screen\_id | int | 11 | Not Null | Screen Id |
| 3 | name | varchar | 40 | Not Null | Name |
| 4 | start\_time | time |  | Not Null | Start Time |
| 5 | charges | int | 11 | Not Null | Ticket Price |

**Table Name:** tblshows

**Purpose:** To store the details of the Shows

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | s\_id | int | 11 | Primary Key | Show Id |
| 2 | st\_id | int | 11 | Not Null | Show Time Id |
| 3 | theatre\_id | int | 11 | Not Null | Theatre Id |
| 4 | movie\_id\_ | int | 11 | Not Null | Movie Id |
| 5 | start\_date | date |  | Not Null | Start Date |

**Table Name:** tbltheatre

**Purpose:** To store the details of the Theatre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S. No.** | **Field Name** | **Data type** | **Size** | **Constraint** | **Description** |
| 1 | id | int | 11 | Primary Key | Id |
| 2 | name | varchar | 30 | Not Null | Theatre Name |
| 3 | address | varchar | 50 | Not Null | Theatre Address |
| 4 | place | varchar | 30 | Not Null | Theatre Place |
| 5 | state | varchar | 20 | Not Null | State |
| 6 | pin | int | 11 | Not Null | Pin Code |

#### USER INTERFACE DESIGN

**Home Page**

Register

Login

Movies

Home

Ticket Booking

Search

Enter a Movie Name

Films in Theatre

Movie Trailers

Upcoming Movies

#### Theatre Assistant Page

**THEATRE ASSISTANT**

View Shows

Today Bookings

Today Shows

Add Shows

View Movies

Add Movie

Home

Theatre Assistance

Wayne Cinemas

#### Admin Page

Admin

OMTBS

Home

Add Theatre

Upcoming Movie News

**User Login:**

New Here ? Register

Login

Password

E-mail

Sign in to Start your Session

Login

Password

#### User Registration:

Register

Name

Age

Select Gender

Mobile Number

Email

Password

Password

Continue

#### Booking Page:

Rs 250

Screen 1

1

10:30 AM Second Show

24-Aug-2024

Wayne Cinemas

Amount

Number of Seats

Show Time

Date

Screen

Theatre

Book Now

* 1. **NORMALIZATION**

Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization divides the larger table into the smaller table and links them using relationship. The normal form is used to reduce redundancy from the database table.

#### First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

* + - It should only have single(atomic) valued attributes/columns
    - Values stored in a column should be of the same domain
    - All the columns in a table should have unique names
    - And the order in which data is stored, does not matter

#### Second Normal Form (2NF)

For a table to be in the Second Normal Form,

* + - It should be in the First Normal form
    - And, it should not have Partial Dependency. Partial Dependency occurs when a non-prime attribute is functionally dependent on part of a candidate key

#### Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

* + - It is in the Second Normal form.
    - And, it doesn't have Transitive Dependency.

#### Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

* + - R must be in 3rd Normal Form
    - For each functional dependency (X → Y), X should be a super Key.

#### Fourth Normal Form (4NF)

A table is said to be in the Fourth Normal Form when,

* + - It is in the Boyce-Codd Normal Form.
    - And, it doesn't have Multi-Valued Dependency.

# SYSTEM TESTING

## SYSTEM TESTING

#### TYPES OF TESTING

Testing is the major quality measure technique employed during software development process. After the coding phase, computer programs are available that can be executed for testing purpose. Testing not only has to uncover errors introduced during coding, but also locates errors committed during the previous phase. Thus, the aim of testing is to uncover requirements, design or coding errors in the program.

The basic types of testing are:

* + - Unit testing
    - Integration testing
    - Validation testing
    - Output testing
    - User Acceptance testing

#### UNIT TESTING

This is the first level of testing. In this different module are tested against the specification produced during the design of the modules. Unit testing is done for the verification of code produced during the coding of single program module in an isolated environment. Unit testing first focuses on the modules independently of one another to locate errors.

#### INTEGRATION TESTING

After the modules are tested individually, they must be tested in combination with each other to be sure that the interfaces are correct. This is known as integration testing. Hence, we consider interfacing of various modules. Thus in the integration testing step, all the errors uncovered are corrected for the next testing steps.

#### VALIDATION TESTING

Validation testing gives the final assurances that the software meets all functional, behavioural and performance requirements. The software is completely assembled as a package. Validation succeeds when the software functions in a manner in which the user expects. Validation refers to the process of using software in a live environment in order to find errors. If the password was given wrongly by customers then it shows the check password error. Then if the username and password are not typed correct then it shows check username and password error. In the field, if the customers type any character other than numbers then it displays a warning message to give only numbers.

#### OUTPUT TESTING

After performing the validation testing the next step is output testing of the proposed system since no system could be useful if it does not produce the required output generated or considered in to two ways, one is on screen and another is printed format. The output format on the screen is found to be correct as the format was designed in the system design phase according to the user needs. If the user gives their correct username and password then it logins to the corresponding page.

#### USER ACCEPTANCE TESTING

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes where required.

#### TYPES OF VALIDATIONS

A Validation control enables to validate an input and display an error message if necessary.

Validation types are given below

#### REQUIRED FIELD VALIDATION

The Required Field Validator is actually very simple, and yet very useful. One can use it to make sure that the user has entered something in a Text Box control. In every form required field validator is assigned to fulfil all the specification.

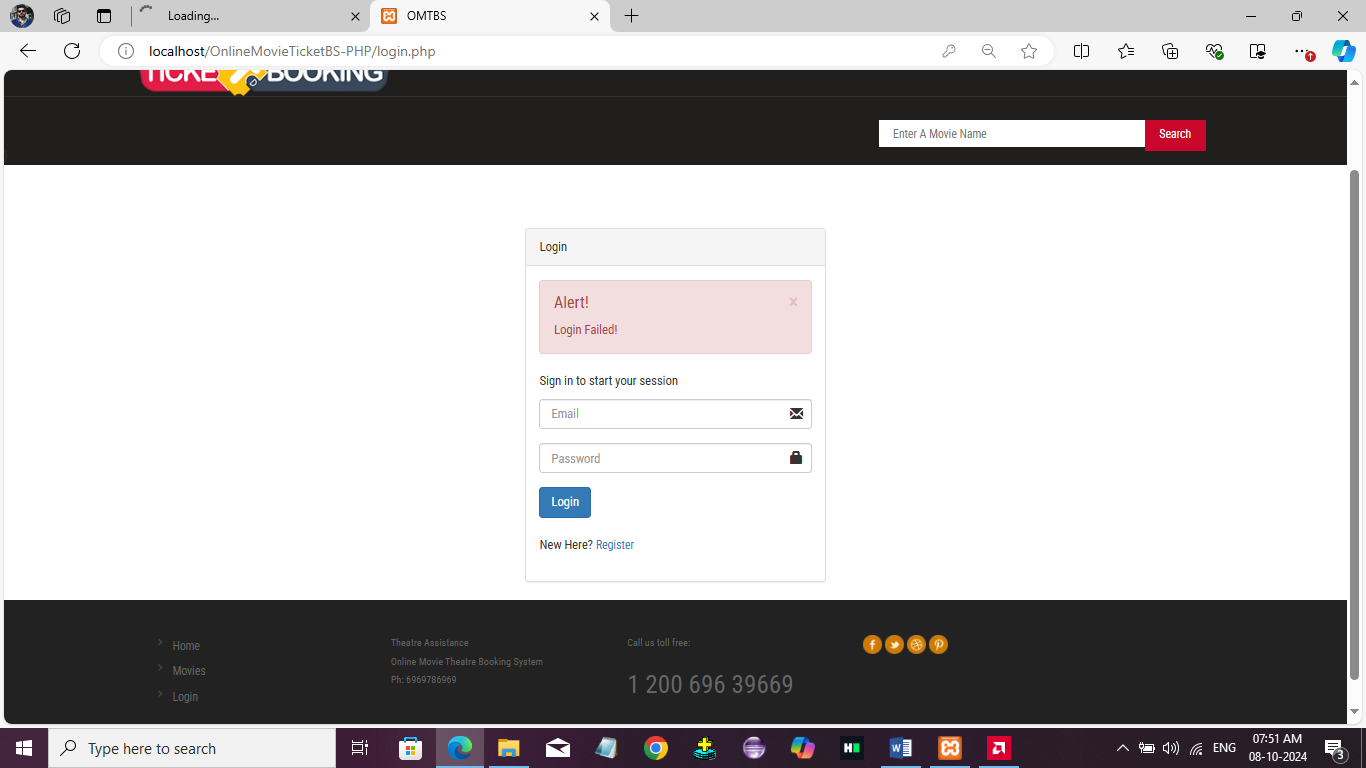
#### REGULAR EXPRESSION VALIDATION

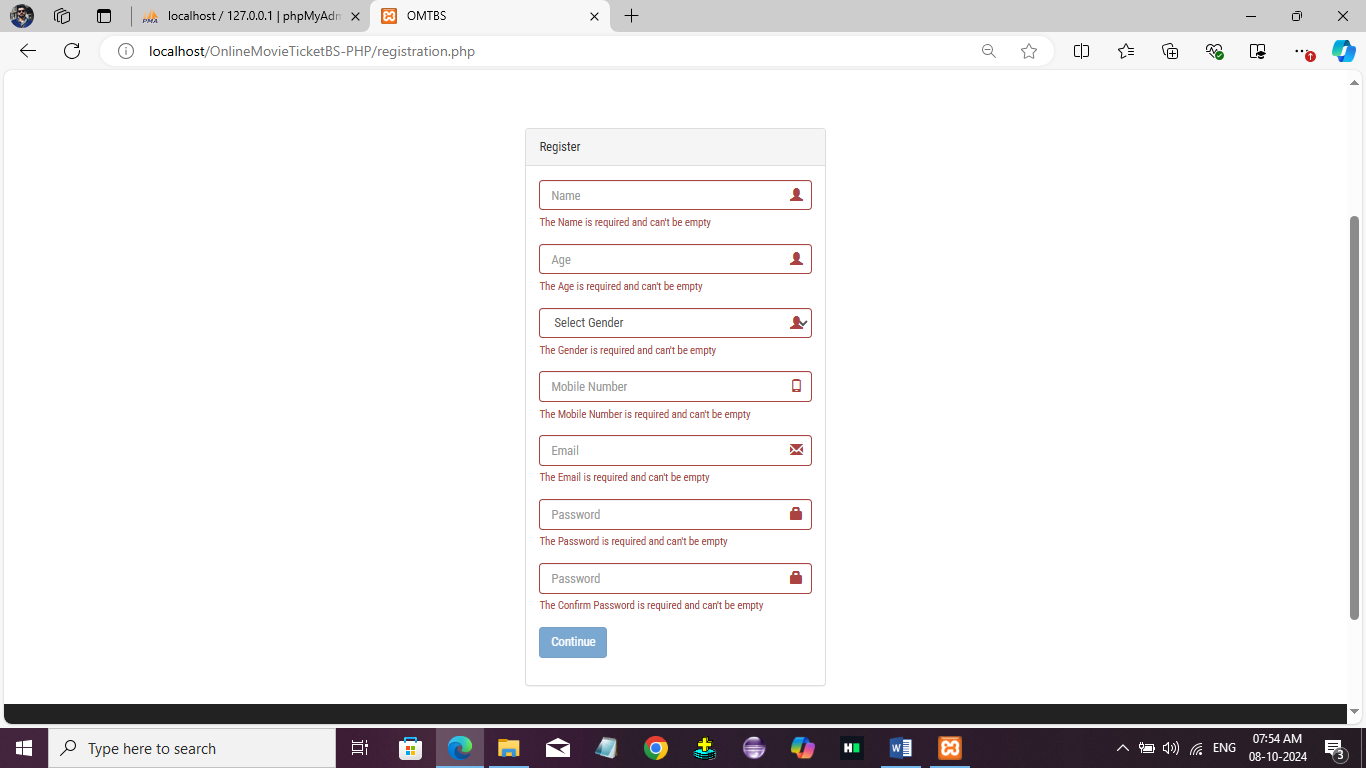
Regular Expression Validator is one of the most useful validators, because it can be used to check the validity of any kind of string. In this project regular expression validator is assigned for email checking entries.

#### RANGE VALIDATION

The Range Validator does exactly what the name implies; it makes sure that the user input is within a specified range. It is used to validate numbers, strings and dates, which can make it useful ina bunch of cases. In this project, range validator is assigned for checking phone numbers.

* 1. **ERROR MESSAGES**





# USER MANUAL

ONLINE MOVIE TICKET BOOKING SYSTEM

## 6 USER MANUAL

* 1. **INSTALLATION MANUAL**
     + Download the required software.
     + Unzip the files.
     + Install all the software one by one.
     + Install MYSQL.
     + Install Apache.
     + Open the web browser and type ‘localhost’ as the address.
     + Unzip the PHP file and copy it to the C disk.
     + Find the file named ‘httpd.conf” at ‘C:
     + Open the htdocs folder at ‘C:
     + Open your web browser and enter ‘localhost/OnlineMovieTicketBS-PHP.’ as the address.
     + If the browser shows the PHP version and other things, it means the PHP is successfully installed.

## OPERATIONAL MANUAL

1. Open the project and go to the home page.
2. Click on the "Login" option in the navigation bar, there you can login as admin.
3. Once you're logged in as an admin, you can manage and add Theatres, Upcoming Movie
4. If you are logged in as a customer you can view the movies posted by different Theatres in candidate dashboard
5. The customers can also book the Tickets by providing relevant details, including payment information and preferences, to enhance their booking experience.
6. The Theatre Assistance login also available in this website. If you are logged in as a Theatre Assistance you manage movies and shows and also can view movies.
7. A separate type of registration page also available for different types of accounts
8. Users can access articles, tips, and advice related to careers in the film and entertainment industry, helping them explore opportunities while enjoying their movie experiences.

Department of Computer Applications, St. Xavier’s College (Autonomous), Palayamkottai Page 24

# SYSTEM IMPLEMENTATION

#### SYSTEM IMPLEMENTATION

* 1. **SPECIAL FEATURES OF THE LANGUAGES PHP**

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus lerdorf in 1994; the PHP reference implementation is now produced by the PHP group. PHP originally stood for personal home page, but it now stands for the recursive initialism. PHP code may be executed with a command line interface (CLI), embedded into HTML code, or it can be used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a common gateway interface executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page.PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP license. PHP has been widely ported and can be deployed on most web servers on almost every operation system and platform, free of charge.

The PHP language evolved without a written formal specification or standard until 2014, with the original implementation acting as the de facto standard which other implementations aimed to follow PHP specification.

#### UNIQUE FEATURES OF PHP

**Simple, Familiar and ease of use:** It is popularly known for its simplicity, familiarity and easy to learn the language as the syntax is similar to that of ‘C’ or Pascal language. So the language is very logical and well organized general-purpose programming language. Even people with a normal programming background can easily understand and capture the use of language. PHP is very advantageous for new users as its a very reliable, fluent, organized, clean, demandable and efficient. The main strength of PHP is the availability of rich pre-defined functions.

**Loosely typed language:** PHP encourages the use of variables without declaring its data types. So this is taken care at the execution time depending on the value assigned to the variable. Even the variable name can be changed dynamically.

**Flexibility:** PHP is known for its flexibility and embedded nature as it can be well integrated with HTML, XML, Java script and many more. PHP can run on multiple operating systems like Windows, Unix, Mac OS, Linux, etc. The PHP scripts can easily run on any device like laptops, mobiles, tablets, and computer. It is very comfortably integrated with various Databases. Desktop applications are created using advanced PHP features. The executable PHP can also be run on command-line as well as directly on the machine. Heavyweight applications can be created without a server or browser. It also acts as an excellent interface with relational databases.

**Cross-platform compatibility:** PHP is multi-platform and known for its portability as it can run on any operating System and windows environments. The most common are XAMPP and LAMP. As PHP is platform-independent, it’s very easy to integrate with various databases and other technologies without re-implementation. It effectively saves a lot of energy, time and money.

**Open Source:** All PHP frameworks are open sources, No payment is required for the users and its completely free. User can just download PHP and start using for their applications or projects. Even in companies, the total cost is reduced for software development providing more reliability and flexibility. It supports a popular range of databases like MySQL, SQLite, Oracle, Sybase, Informix, and PostgreSQL

**Error reporting and exceptions:** PHP supports more errors reporting constants to generate errors and relevant warning at runtime. For example**,** E\_ERROR, E\_WARNING, E\_PARSE, E\_STRICT**.**

**Active community support:** PHP is very rich with many diverse online community developers to help beginners for web-based applications. These worldwide volunteers contribute many features as well as new versions for PHP libraries. Even they contribute a translation in different languages to help out programmers. There is a bundle of third-party open-source libraries which provide basic functionalities. Even the documentation given by the official site helps in implementing new features providing access to a variety of creative imagination.

**Maintenance:** When dealing with big projects, maintenance of code is also an important aspectof the web development process. There are many PHP frameworks for example MVC (Model View Controller) which makes development and maintenance of code easier. Files belonging to the different module are maintained separately.

**PDO Class:** PHP Data Objects are created by PDO class which gives a good abstraction layer for database drivers. The PDO Classes are enriched with functions which are database independent. It means the same functions are used for similar actions for different databases without re- development as long as it supports PDO. In this way, the application becomes more portable saving lot of time and effort. Use of PDO helps the application from SQL injection attacks.

**Memory and CPU usage information:** PHP can provide memory usage information from functions like memory\_get\_usage() or memory\_get\_peak\_usage(), which can help the developers optimize their code. In the similar way, the CPU power consumed by any script can be retrievedfor further optimization.

**Object oriented features:** PHP supports object-oriented programming features, resulting in increased speed and introducing added features like data encapsulation and inheritance at many levels.

#### MySQL

MySql is an open source relational database management system (RDMBS). It’s name is a combination of “My”, the name of co-founder Michael Widenius’s daughter, and “SQL”, the abbreviation for structured Query Language.

MySql is free and open-source software under the terms of the GNU general public license, and is also available under a variety of proprietary licenses. MySql was owned and sponsored by the Swedish company MySql AB, which was bought by sun Microsystems. In 2010 when oracle acquired sun, Widenius forked the open-source MySql project to create MariaDB.

MySql is a component of the lamp web application software stack, which is an acronym for Linux, Apache, MySql, and Perl. MySql is used by many database-driven web applications, including drupal, joomla, phpBB, and Word Press. MySql is also used by many popular websites, including Google, face book, Twitter and You Tube. The main features of MySql includes

**Easy to use**: MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

**It is secure**: MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL

**Client/ Server Architecture**: MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

**Free to download**: MySQL is free to use so that we can download it from MySQL official website without any cost.

**Compatible on many operating systems**: MySQL is compatible to run on many operating systems, like Novell NetWare, Windows\* Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

# FUTURE ENHANCEMENT

ONLINE MOVIE TICKET BOOKING SYSTEM

## FUTURE ENHANCEMENT

### The project “Online Movie Ticket Booking System” can have further updates. Users can receive personalized movie recommendations based on their viewing history and preferences. The locations of movie theaters can be shared using Google Maps to help users find directions easily.

### Additionally, booking confirmations and reminders can be sent through email to the corresponding user, ensuring they stay informed about their ticket status

Department of Computer Applications, St. Xavier’s College (Autonomous), Palayamkottai

Page 30

# CONCLUSION

ONLINE MOVIE TICKET BOOKING SYSTEM

## CONCLUSION

In conclusion, the "Online Movie Ticket Booking System" is an innovative project developed using PHP, JavaScript, and MySQL technologies that simplifies the process of booking movie tickets in an efficient manner. The platform offers a comprehensive solution for users to explore and purchase tickets for various films.

In the existing ticket booking systems, much of the work is done manually, leading to delays and inefficiencies. A significant challenge is the difficulty in finding available showtimes and seats, along with the lack of timely notifications about bookings. This project addresses these issues by providing a user-friendly interface for customers to easily book tickets and receive real-time updates.

The project focuses on security and scalability, ensuring data privacy and protection. Overall, this system offers a convenient and efficient way for users to book movie tickets, manage their bookings, and enjoy a seamless viewing experience.

Department of Computer Applications, St. Xavier’s College (Autonomous), Palayamkottai

Page 32

# BIBLIOGRAPHY

ONLINE MOVIE TICKET BOOKING SYSTEM

#### BIBLIOGRAPHY

**BOOK REFERENCES**

* + - Robin Nixon, “Learning **PHP**, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites”, Second Edition, O’Reilly Publications, 2012
    - Thomas Powell, “The Complete Reference: HTML and XHTML”, Fifth Edition, Tata McGraw Hill Publication, 2010
    - Paul Hudson, “PHP in a Nutshell”, O’Reilly Publications, 2005
    - Roger S. Pressman, “Software Engineering: A Practitioner Approach”, Eighth Edition, McGraw Hill, 2015
    - William E, Perry, “Effective Methods for Software Testing”, Third Edition, John Wiley, 2015

#### WEBSITE REFERENCES

* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.stackoverflow.com](http://www.stackoverflow.com/)
* [www.w3schools.com](http://www.w3schools.com/)

Department of Computer Applications, St. Xavier’s College (Autonomous), Palayamkottai

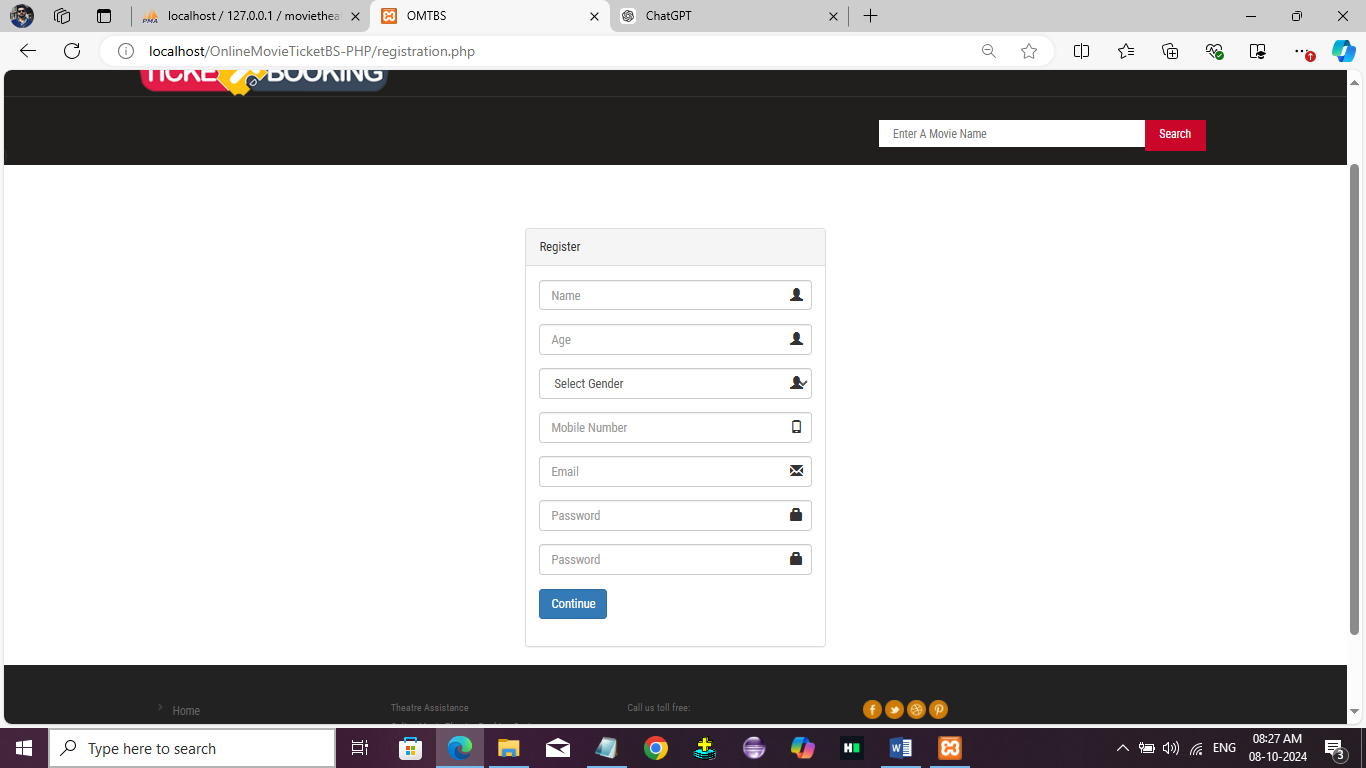
Page 34

# APPENDIX

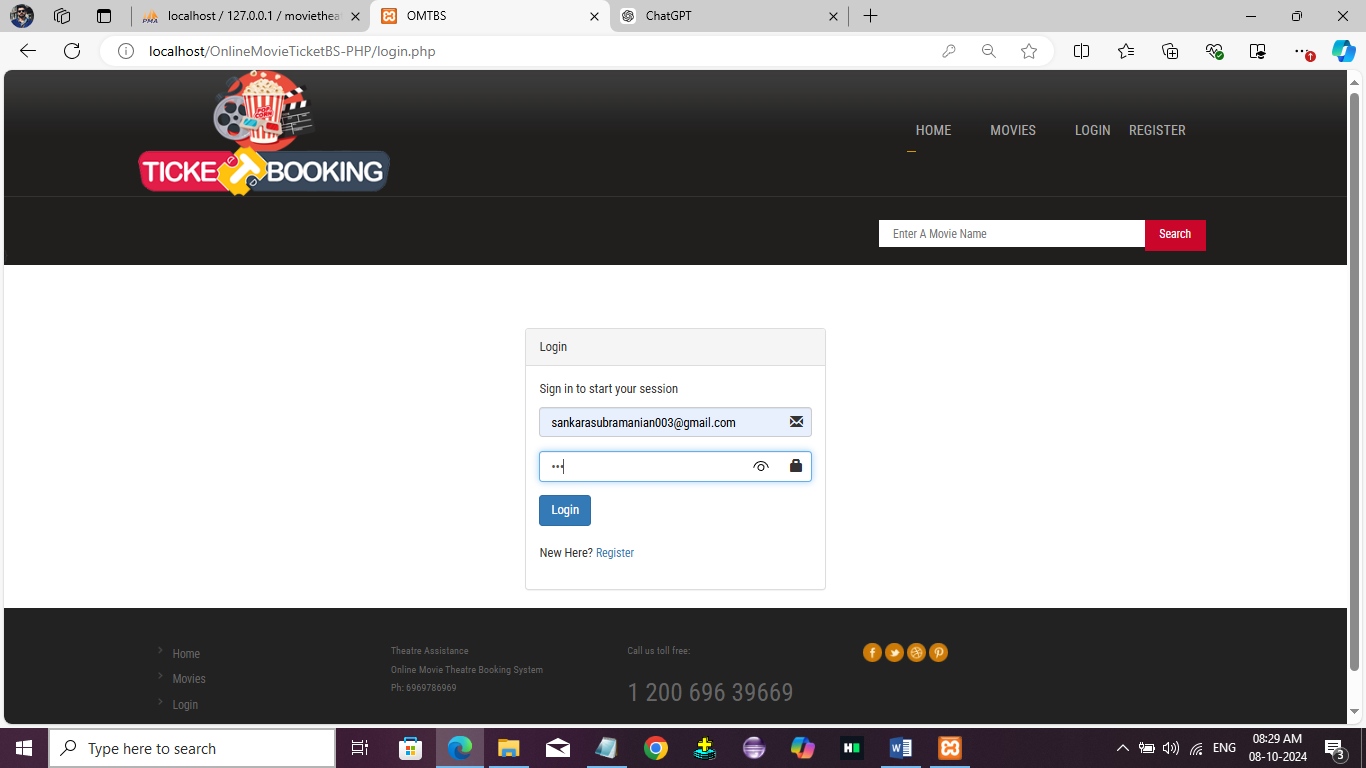
## APPENDIX

* 1. **SAMPLE SCREEN LAYOUTS**

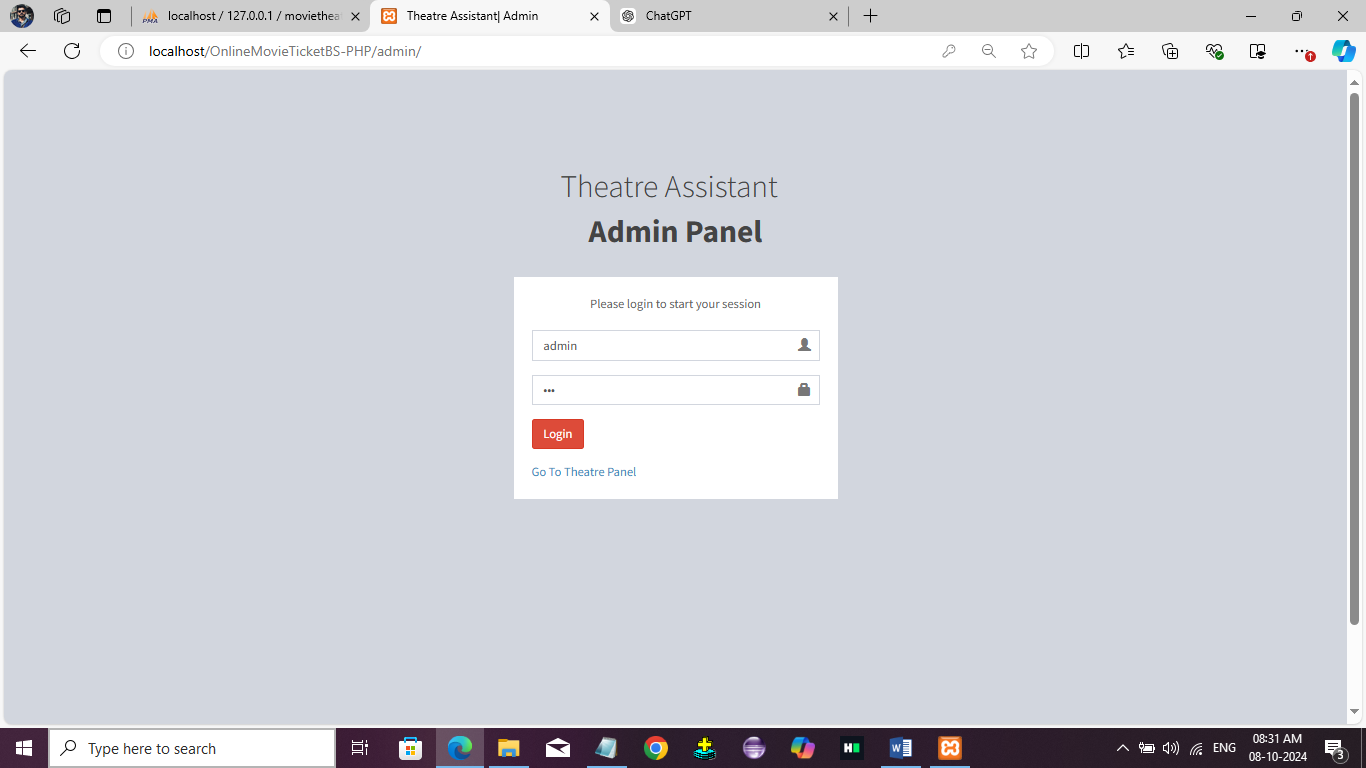
#### Customer Registration



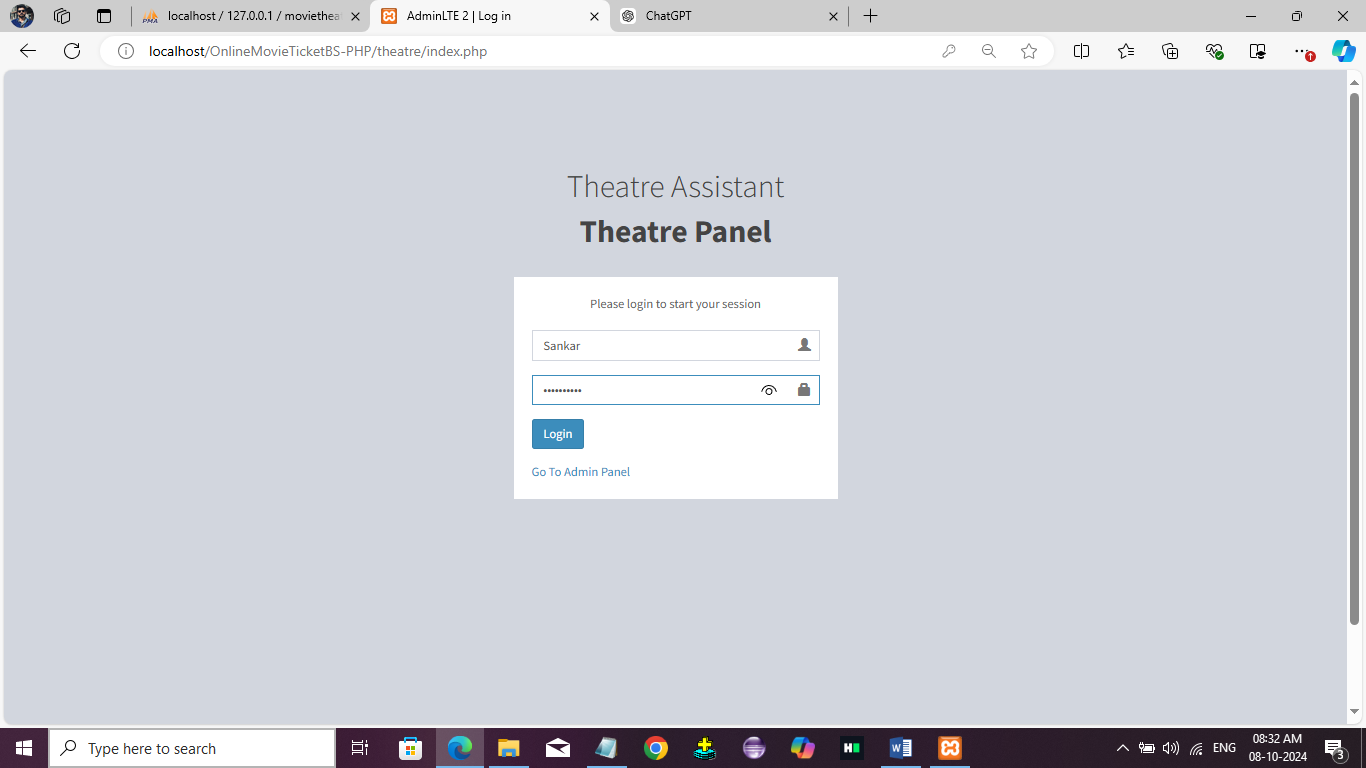
**Login Page**



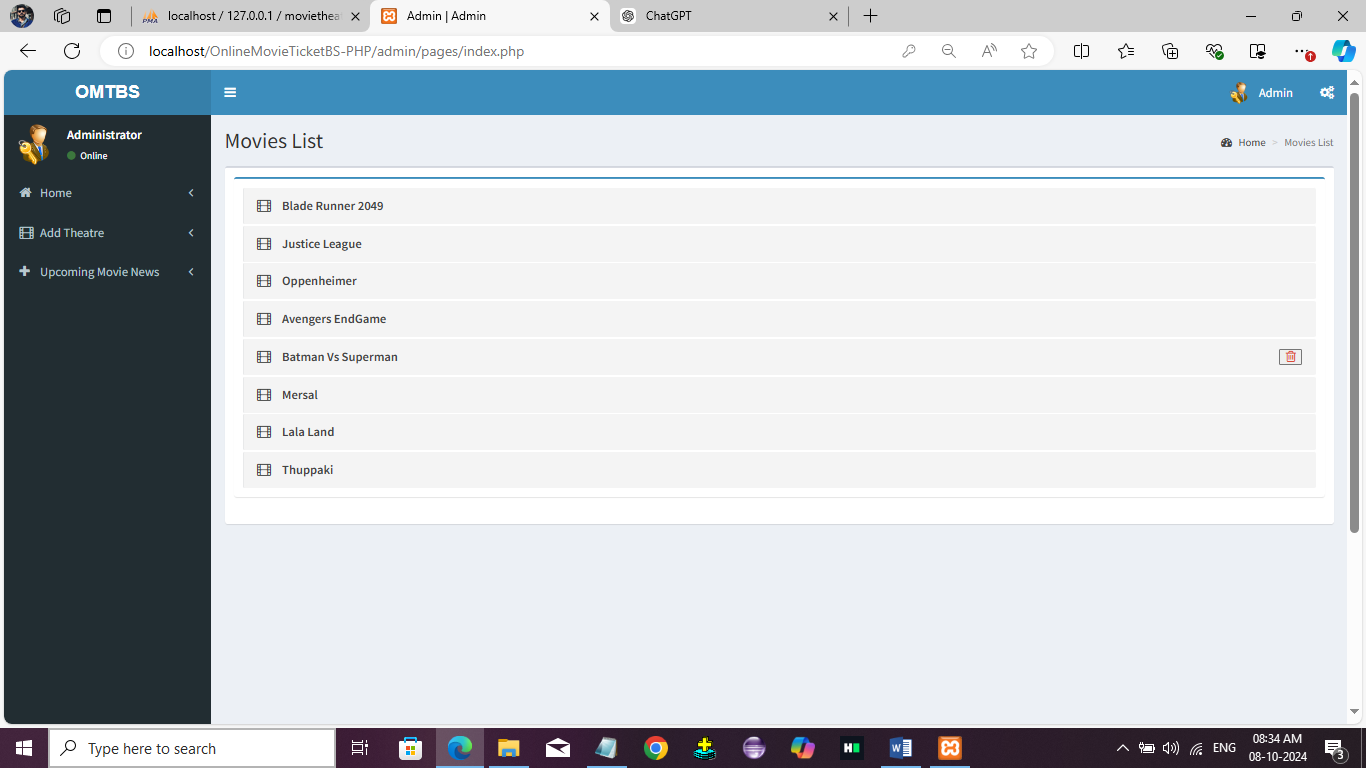
#### Admin Login

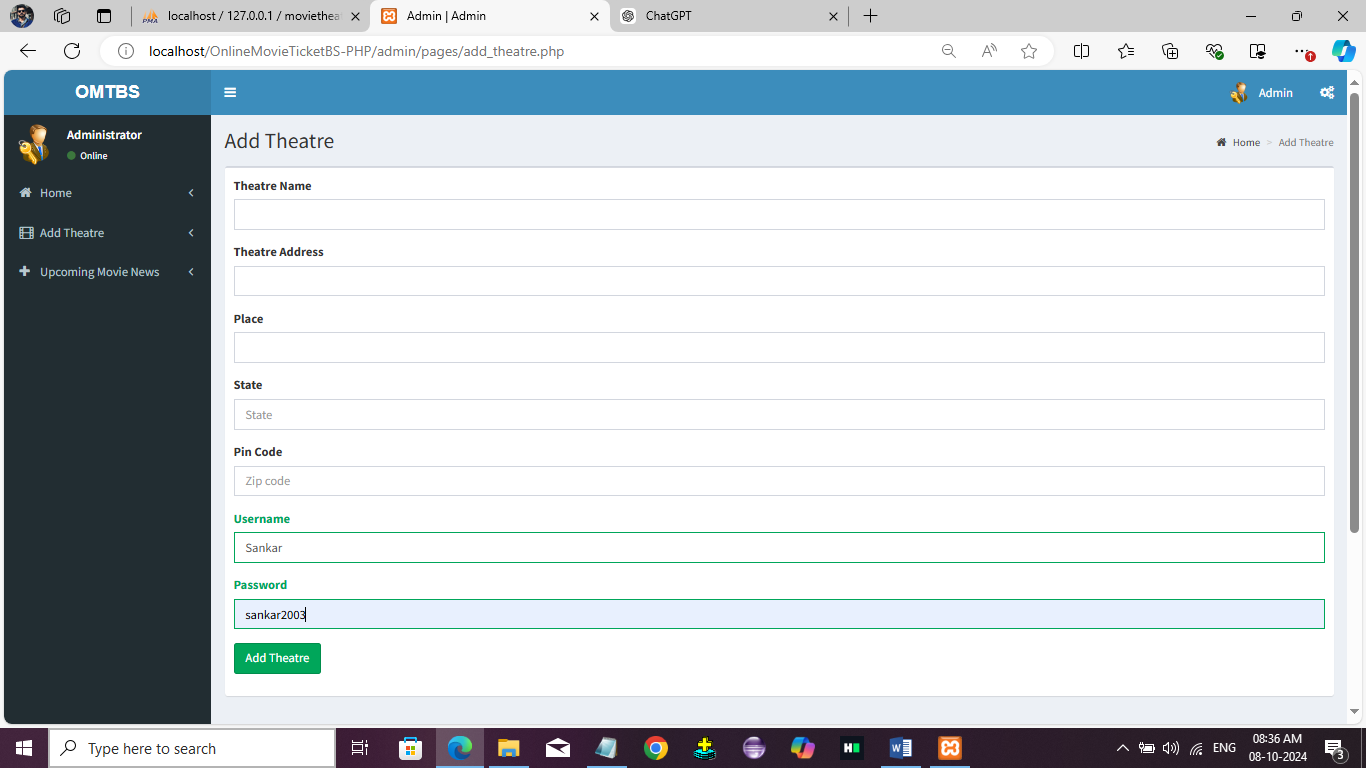


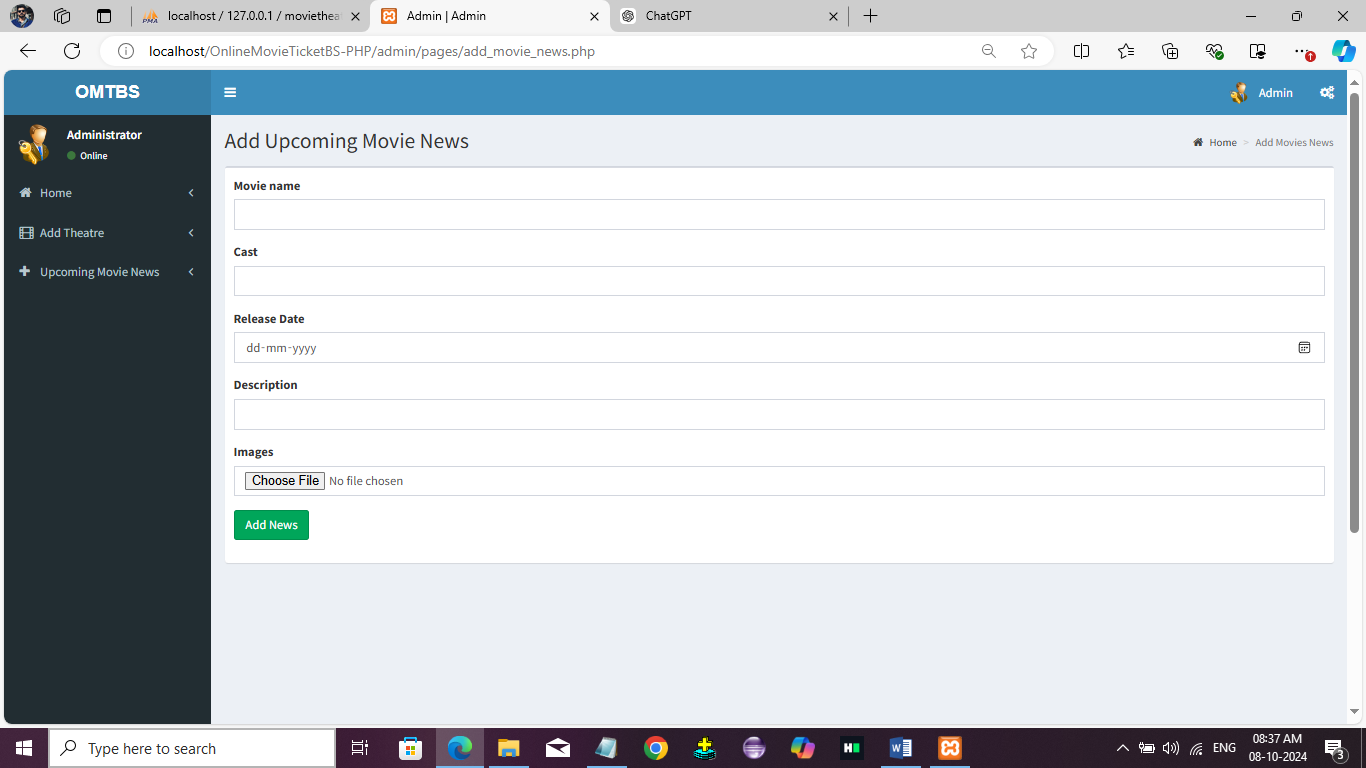
**Theatre Login**



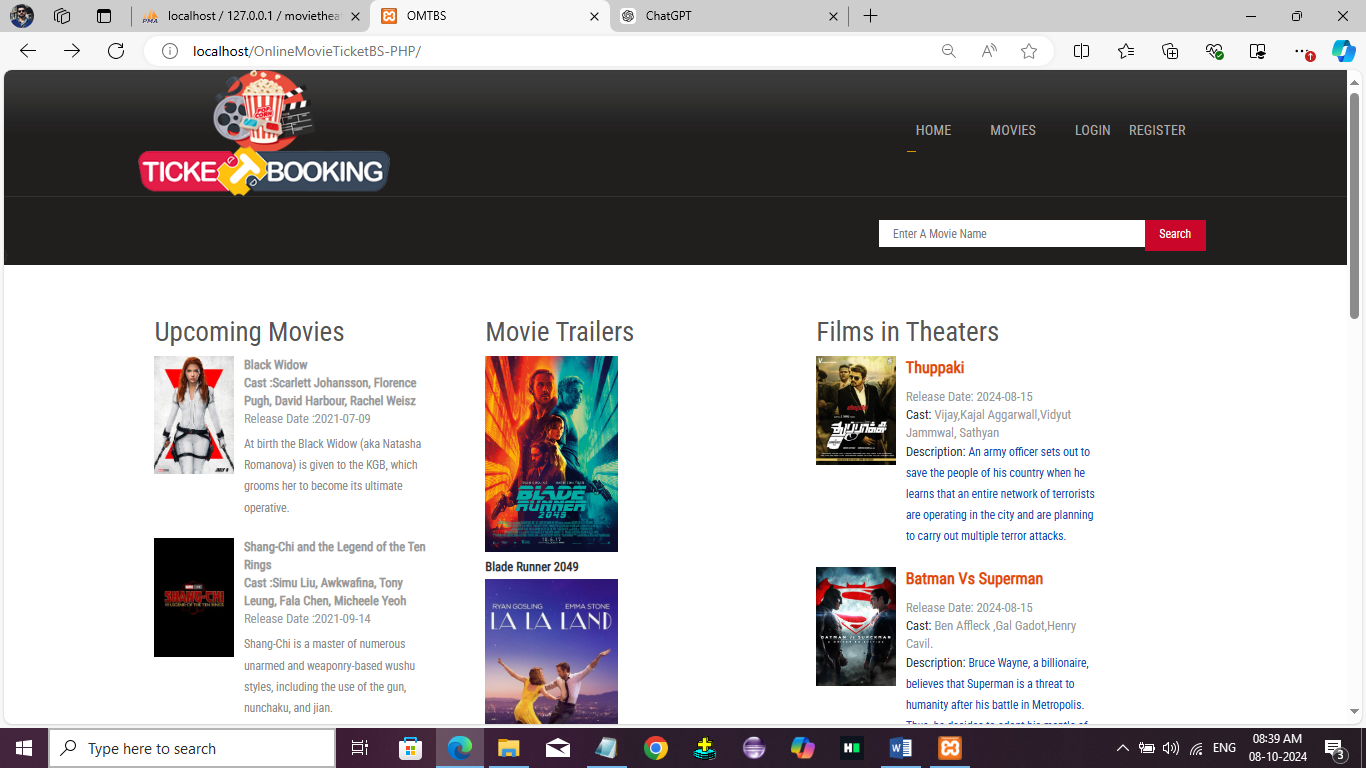
#### Admin Page



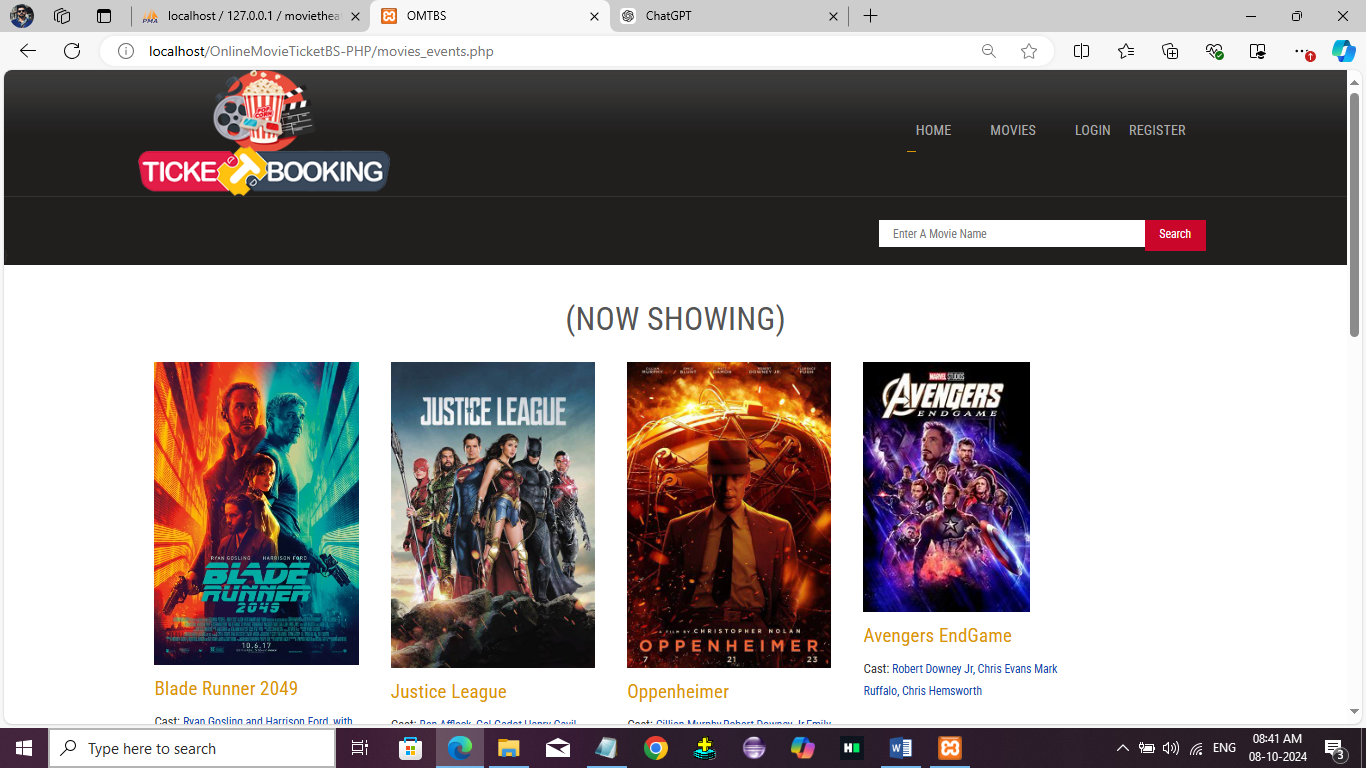


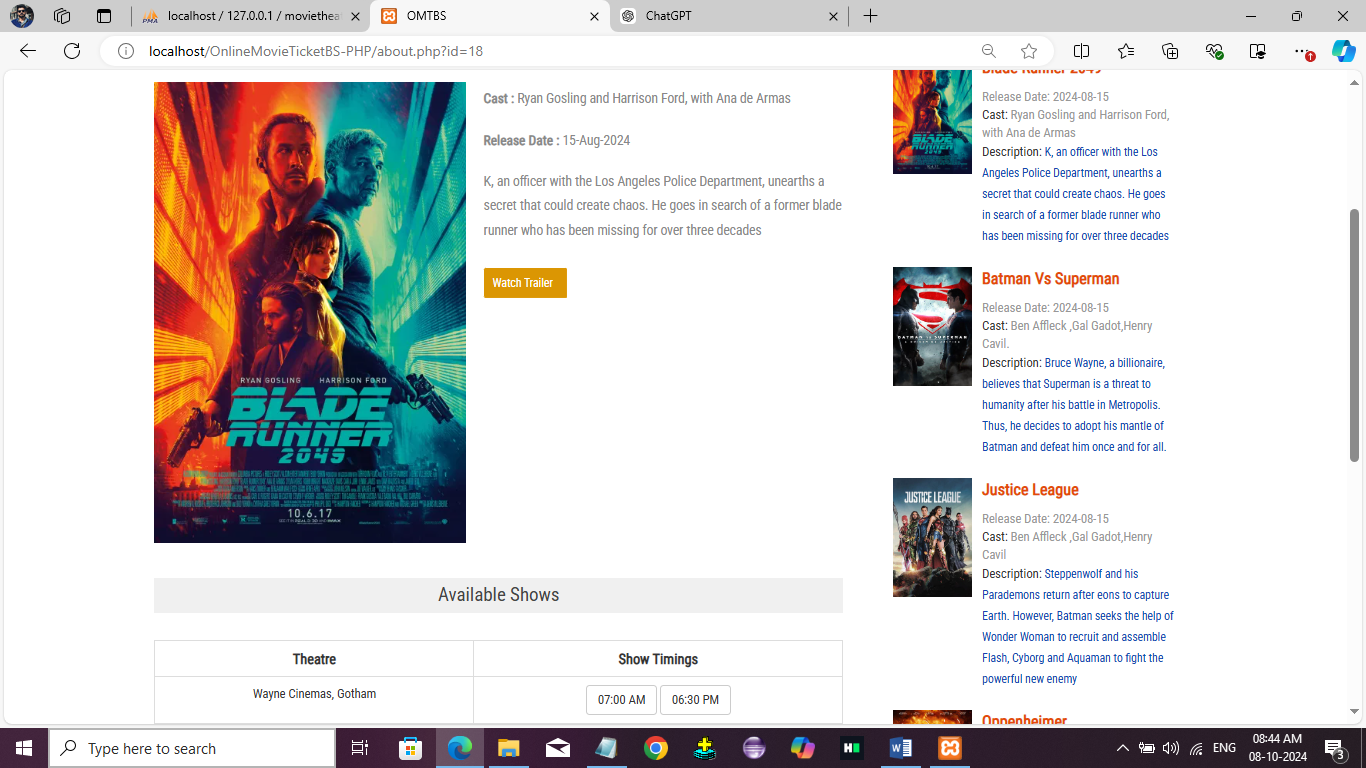


**Home Page**

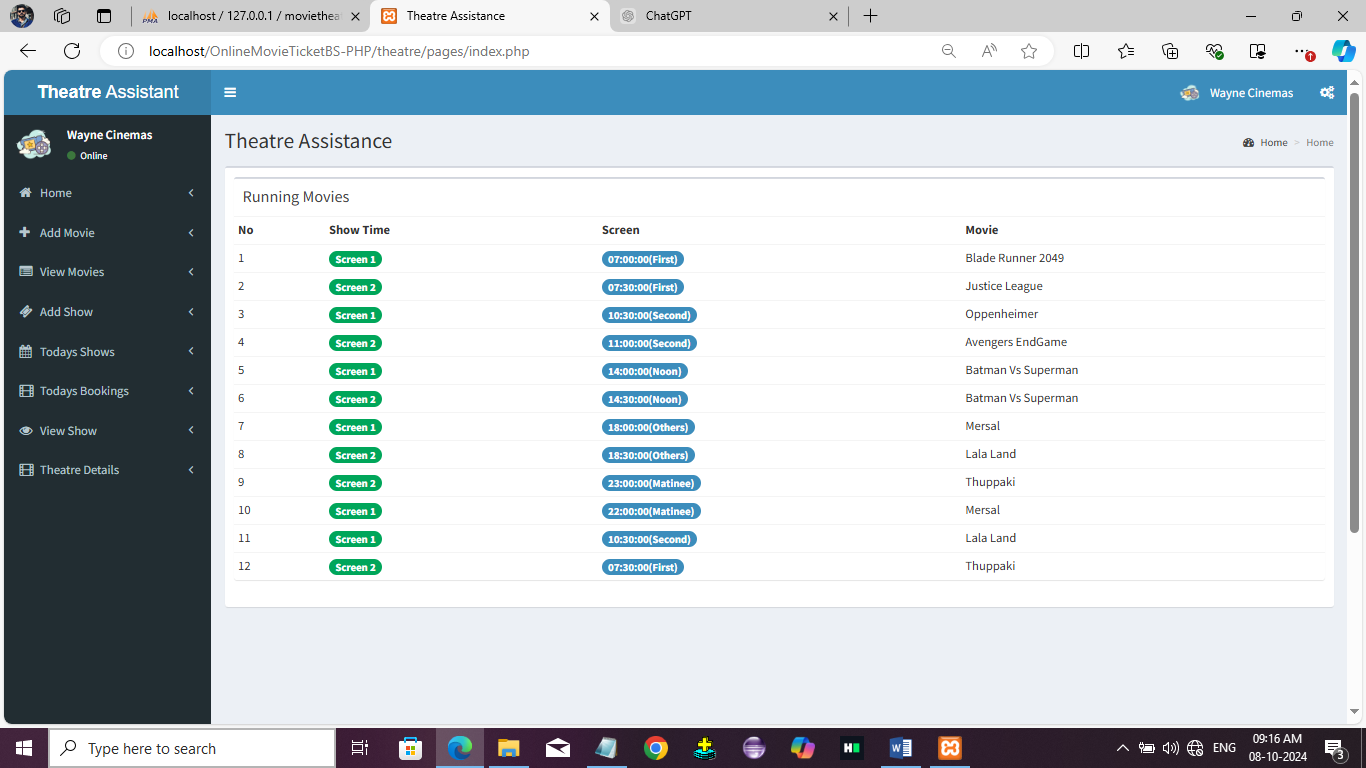


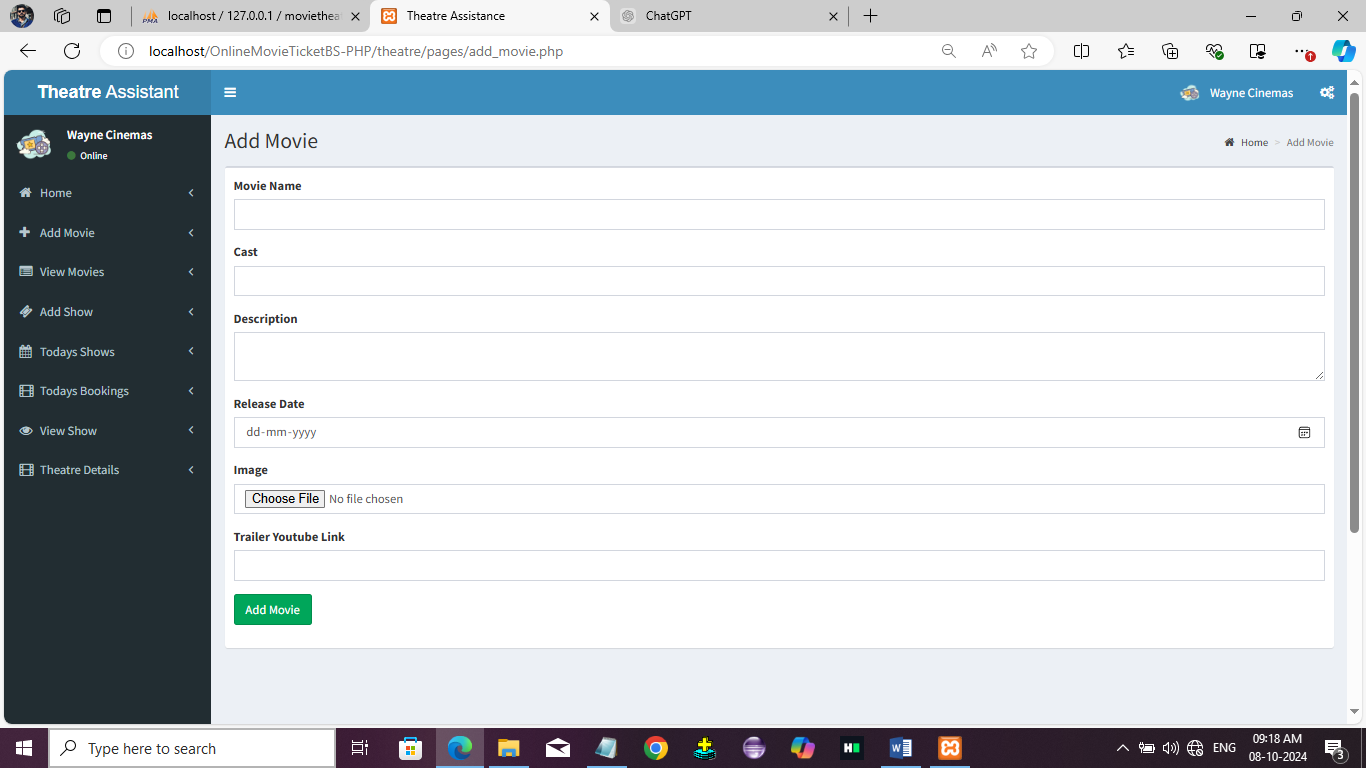
#### Movies Page

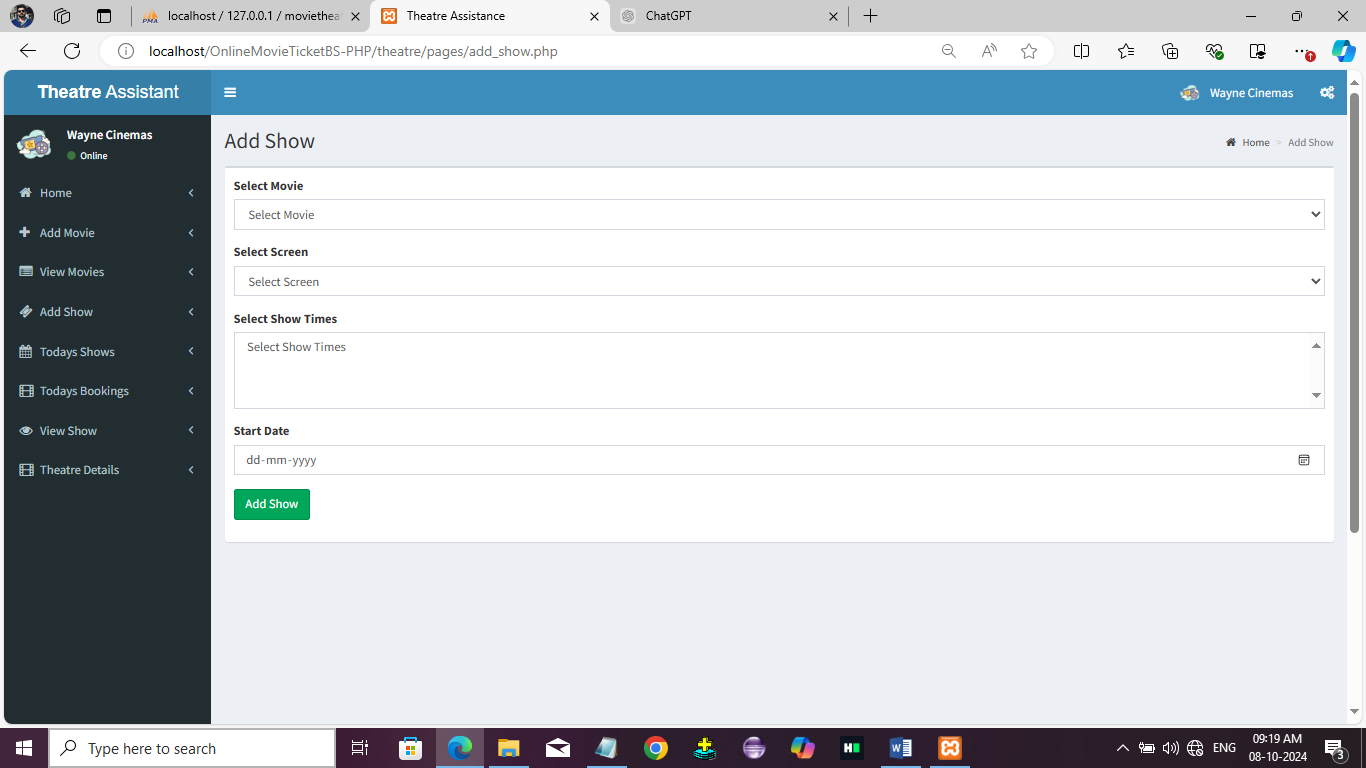


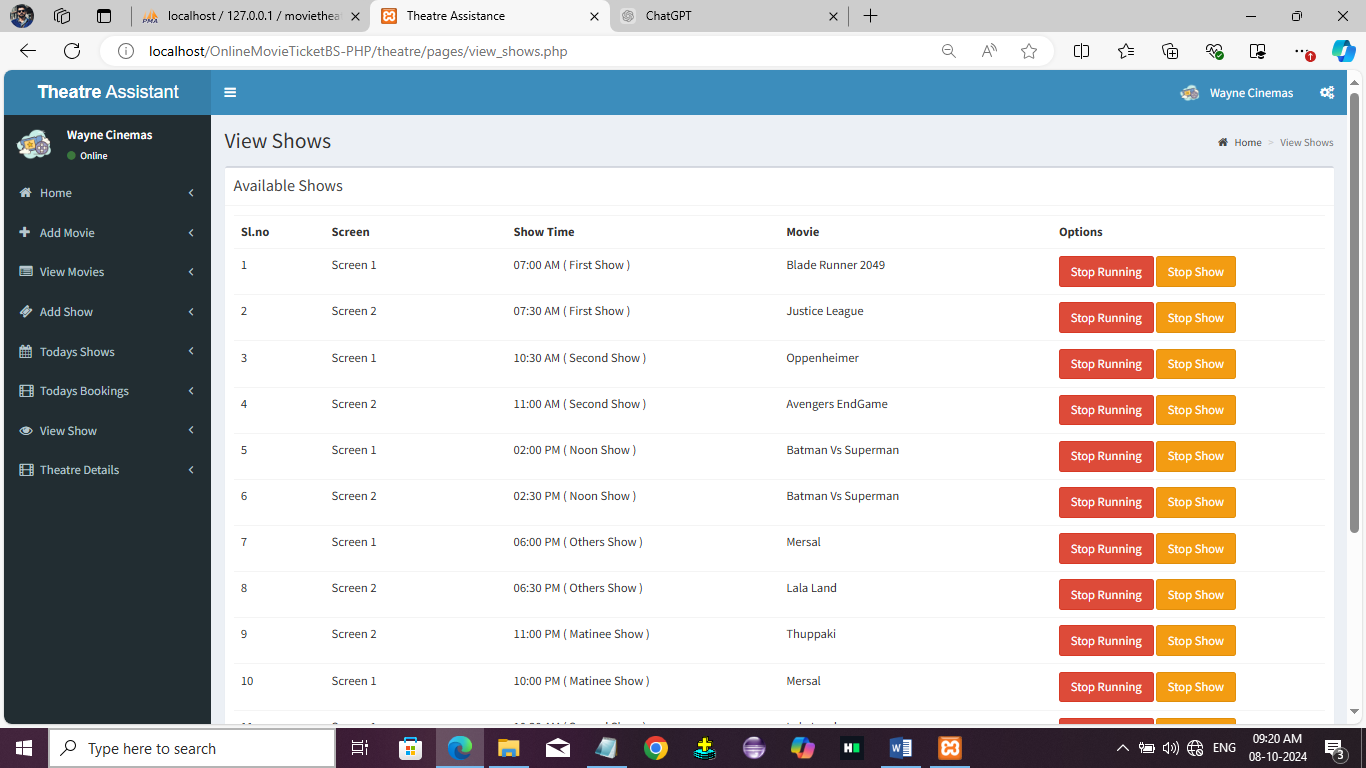


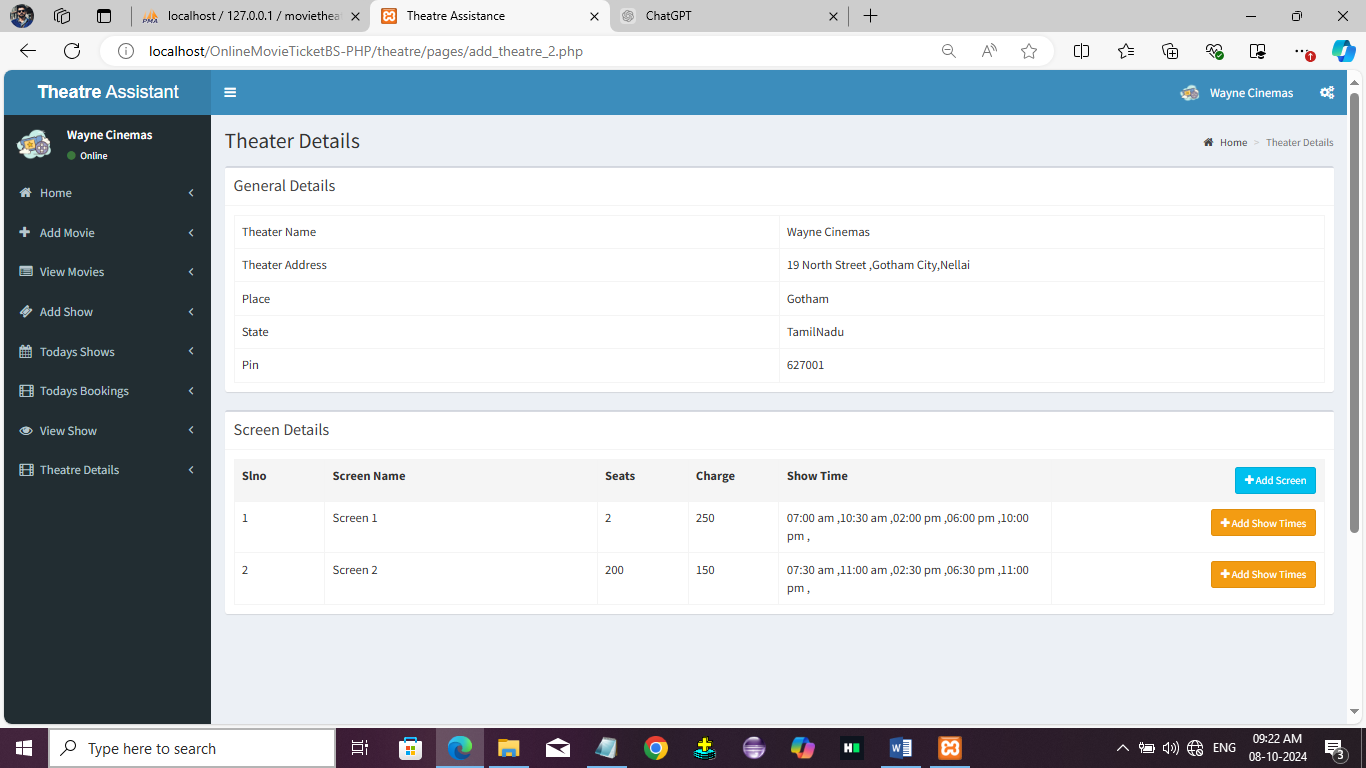
**Theatre Assistance Page**



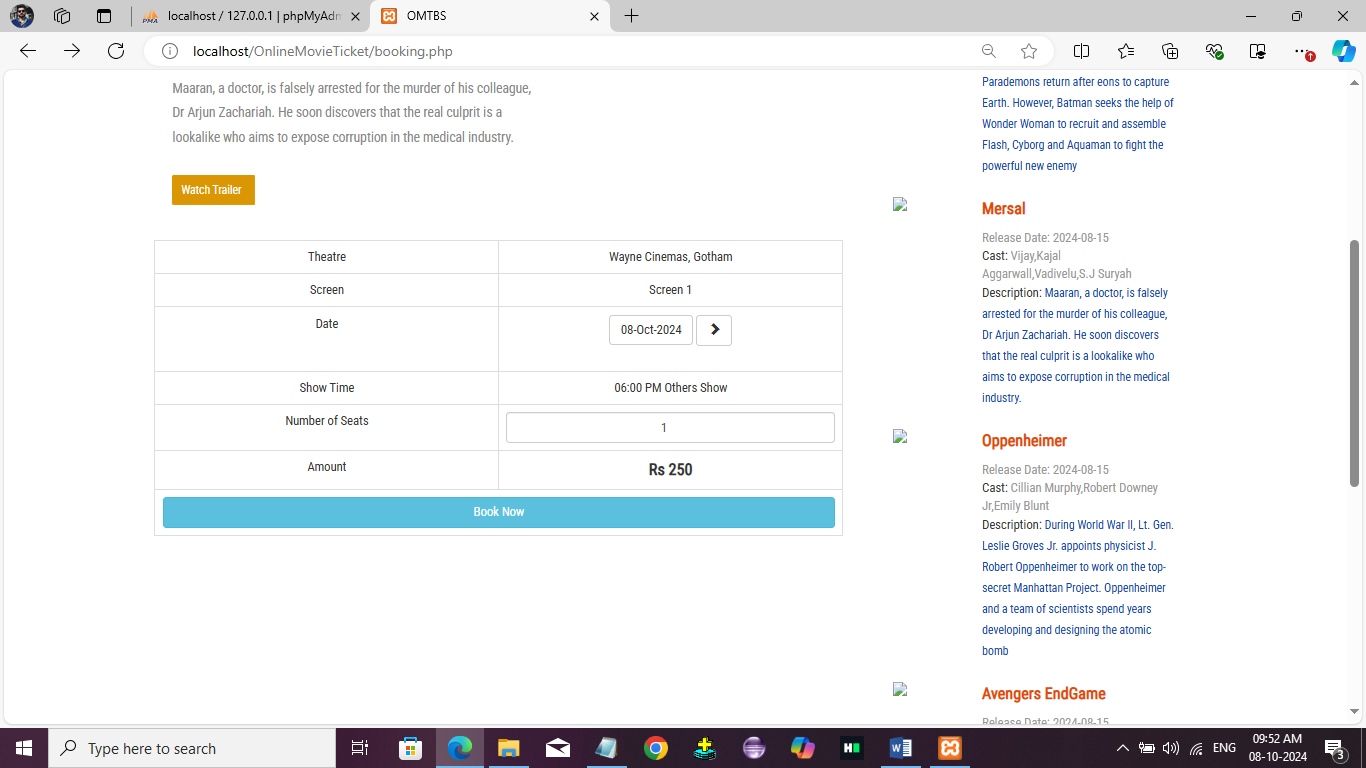




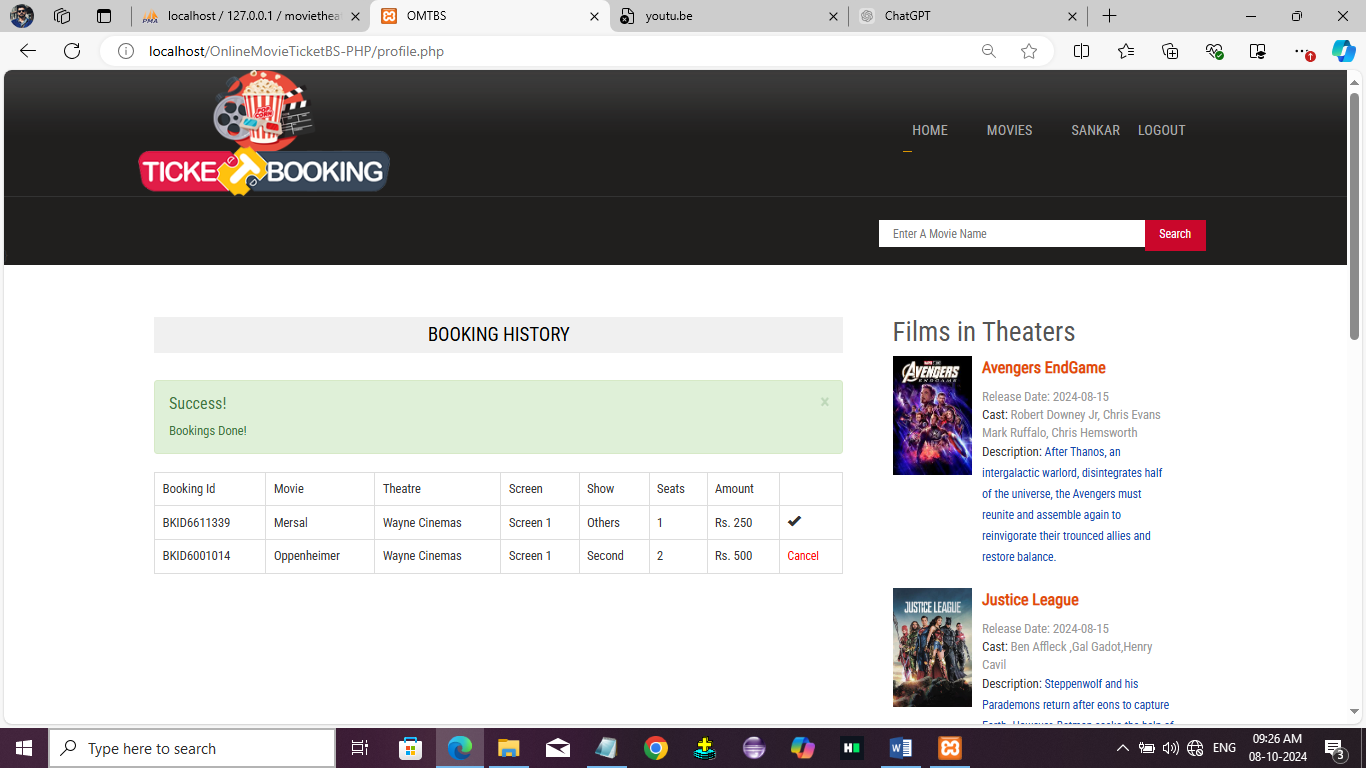




**Booking Page**



**Booking History**



## SAMPLE CODINGS

#### Registration Page:

<?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"));

<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"));

<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"));>

<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"));

<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"));

<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"));

<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:

<?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-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>