AN

(E- COMMERCE PROJECT)

ON

***REAL CINEMAS***

SUBMITTED BY

GROUP NO :- 04

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR. NO | ENROLMENT NO | NAME | IDNO | EXAM NO |
| **1** | **19BCA04174** | **HINGLADIYA MADHVI GANESHBHAI** | **19089** |  |
| **2** | **19BCA04231** | **PIYUSH BAIRAGI** | **19128** |  |
| **3** | **19BCA04260** | **PRAJAPATI RISHANK RAJESHBHAI** | **19132** |  |
| **4** | **19BCA04339** | **TARAPARA VISHVABANE KALPESHBHAI** | **19144** |  |
| **5** | **19BCA04340** | **CHAUHAN VIVEK BHARATBHAI** | **19081** |  |

SUBMITTED TO  
B P COLLEGE OF COMPUTER STDUIES

GANDHINAGAR

OR

S V INSTITUTE OF COMPUTER STUDIES

GANDHINAGAR

Certificate

I here by certify that following group members

1. HINGLADIYA MADHVI GANESHBHAI
2. PIYUSH BAIRAGI
3. PRAJAPATI RISHANK RAJESHBHAI
4. TARAPARA VISHVABANE KALPESHBHAI
5. CHAUHAN VIVEK BHARATBHAI

Have completed the(E-commerce Project) on REAL CINEMAS. E-commerce Technology –VI (BCA -606(B)) subject available at BCA Semester – VI and It is part of syllabus, approved by KadiSarvaVishwavidyalaya, Gandhinagar with the specialization in E-Commerce.

Place:

Date: (Project in charge

INDEX

* Introduction
* Proposed system
* Algorithm of proposed system
* Data flow diagram of Proposed system( 0 / 1/2 level)
* Data Dictionary
* E-R Diagram
* Technology selected for Proposed system
  + Front End
  + Back End
* System design (Screen short after system implementation)

***INTRODUCTION***

* With the online event booking system the customer can book the event ticket online.
* Customer can book or buy movie tickets online where the app lest you to check movie show timings, list of venues , read reviews of customer and book ticket as your choice.

1. **PROPOSED SYSTEM**

* The proposed system have mainly 3 entity :

1. Customer
2. Event organizer
3. System admin
4. **Customer**

* customer can search the event for particular data time & place.
* Customer can select the event.
* Customer can request for ticket.
* Customer can pay payment through online payment.
* Customer can give feedback.

1. **Event organizer :**

* Event organizer can register his event.
* Event organizer uploads like event name ,seat price ,seat numbers & seat type ,place etc.
* Event organizer can conform request of ticket of customer.
* Event organizer can receive the payment & receive the feedback.

1. **System admin :**

* system admin manages events information of events.
* System admin receive payment from customer and send payment to event organizer.
* System admin work as inter mediator for customer and event organizer & manages requests and information of booking.

**Algorithm of proposed system**

**Introduction :**

Real ticket booking system is basically made for providing the customers an anytime and anywhere service for booking cinema tickets and providing information about the movies and their schedule online.

**Algorithm step :**

* Algorithm for customer in proposed system :

step 1 : start

step 2 : customer has login or signup.

step 3 : customer can search event date , time & place.

step 4 : customer select event.

step 5 : customer send request of booking.

step 6: customer receive booking conformation and sends

online payment.

step 7 : customer receive the conformation of payment by e-

mail and text message.

step 8 : customer send feedback.

step 9 : stop

**Algorithm for Event organizer in proposed system:**

Step 1: start

Step 2: event organizer contacts the system.

Step 3: event organizer sends request to system for selling online ticket.

Step 4: event organizer receive conformation of ticket selling and receive estimation of par ticket price.

Step 5: event organizer received request of booking.

Step 6: event organize send payment request to customer.

Step 7: event organizer receive the payment.

Step 8: event organizer receive the feedback.

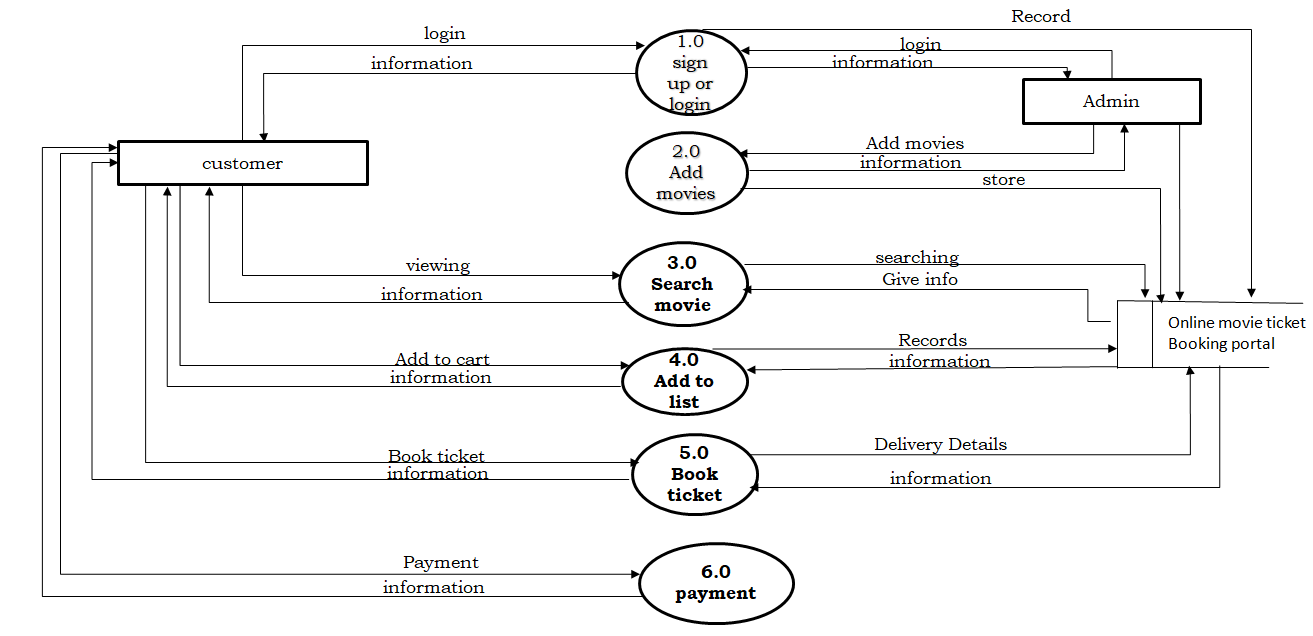
Step 9: stop

**PRPOSED SYSTEM DATA FOLW DIAGRAM (DFD)**

**OF PROPOSED SYSTEM**

**0 LEVEL (CONTEXT LEVEL)**

**1st LEVEL DFD**



**DATA DICTIONARY**

1. **registration table :**

The Purpose of this table is to store the data of the customer.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | User\_id | Int | 11 | This is user ID | Primary key |
| 2 | name | Varchar | 50 | This is customer name | Not null |
| 3 | email | Varchar | 50 | This is email name | Not null |
| 4 | phone | Varchar | 12 | This is phone | Not null |
| 5 | age | Int | 2 | This is age | Not null |
| 6 | gender | Varchar | 10 | This is gender | Not null |

1. **Login Table :**

The Purpose of this table is to store the data of the Login.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | id | Int | 11 | This is Login id | Primary key |
| 2 | User\_id | Int | 11 | This is user id | Not null |
| 3 | username | Varchar | 50 | This is user name | Not null |
| 4 | Password | Varchar | 50 | This is password | Not null |
| 5 | User\_type | Int | 1 | This is user type | Not null |

1. **Movie table :**

This table stores the movie data.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | movie\_id | Int | 11 | This is movie id | Primary key |
| 2 | t\_id | Int | 11 | This is theatre id | Not null |
| 3 | Movie\_name | Varchar | 255 | This is movie name | Not null |
| 4 | cast | Varchar | 500 | This is booking date | Not null |
| 5 | desc | varchar | 1000 | This is booking show date | Not null |
| 6 | release\_date | date |  | This is movie release date | Not null |
| 7 | image | Varchar | 200 | This is movie image | Not null |
| 8 | Video\_url | Varchar | 200 | This is video url | Not null |
| 9 | status | Int | 1 | This is starus | Not null |

1. **Screen details table :**

This table Screen details table.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | Screen\_id | Int | 11 | This is screen id | Primary key |
| 2 | t\_id | Int | 11 | This is theatre  id | Not null |
| 3 | Screen\_name | varchar | 110 | This is screen name | Not null |
| 4 | seats | Int | 11 | This is seats number | Not null |
| 5 | charge | Int | 11 | This is charge | Not null |

1. **Shows table :**

This table stores the shows details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | s\_id | Int | 11 | This is show id | Primary key |
| 2 | st\_id | Int | 11 | This is show time id | Not null |
| 3 | theatre\_id | Int | 11 | This is theatre id | Not null |
| 4 | movie\_id | Int | 11 | This is movie id | Not null |
| 5 | start\_date | Date |  | This is start date | Not null |
| 6 | status | Int | 11 | This is Show available | Not null |
| 7 | r\_status | Int | 11 | This is status | Not null |

1. **Show time table :**

This table stores show time details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Field Name | Data Type | Field size | Description | P/F |
| 1 | st\_id | Int | 11 | This is show time id | Primary key |
| 2 | screen\_id | Int | 11 | This is screen id | Not null |
| 3 | name | Varchar | 40 | This is name | Not null |
| 4 | start\_time | Time |  | This is start time | Not null |

1. **theatre table:**

This table stores theater details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | F1ield Name | Data Type | Field size | Description | P/F |
| 1 | Id | Int | 11 | This is id | Primary key |
| 2 | Name | Varchar | 100 | This is theatre name | Not null |
| 3 | address | Varchar | 100 | This is address | Not null |
| 4 | Place | Varchar | 100 | This is place | Not null |
| 5 | State | Varchar | 50 | This is state | Not null |
| 6 | Pin | Int | 11 | This is pin | Not null |

1. **booking table:**

This table stores booking details.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | F1ield Name | Data Type | Field size | Description | P/F |
| 1 | book\_id | Int | 11 | This is book id | Primary key |
| 2 | ticket\_id | Varchar | 30 | This is ticket id | Not null |
| 3 | t\_id | Int | 11 | This is theatre id | Not null |
| 4 | user\_id | Int | 11 | This is user id | Not null |
| 5 | show\_id | Int | 11 | This is show id | Not null |
| 6 | screen\_id | Int | 11 | This is screen id | Not null |
| 7 | no\_seats | Int | 3 | This is no seats | Not null |
| 8 | amount | Int | 5 | This is amount | Not null |
| 9 | ticket\_date | Date |  | This is ticket date | Not null |
| 10 | date | Date |  | This is date | Not null |
| 11 | status | Int | 1 | This is status | Not null |

**ERD :**

**FRONT-END**

**HTML**

HTML (Hypertext Markup Language) is the code that is used **to structure a web page and its content**. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

## **Features of HTML:**

HTML is the most common used language to write web pages. It has recently gained popularity due to its advantages such as: -

1. It is the language which can be easily understood and can be modified.

2. Effective presentations can be made with the HTML with the help of its allformatting tags.

3. It provides the more flexible way to deign web pages along with the text.

4. Links can also be added to the web pages so it helps the readers to browse the information of their interest.

5. You can display HTML documents on any platforms such as Macintosh, Windows and Linux etc.

6. Graphics, videos and sounds can also be added to the web pages which give an extra attractive look to your web pages.

**PHP**

* + - * PHP is a general-purpose scripting language geared towards web development. It was originally created by Danish-Canadian programmer RasmusLerdorf 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: Hypertext Preprocessor.
      * The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

**Features of PHP**

**Performance:**

* + - * PHP script is executed much faster than those scripts which are written in other languages such as JSP and ASP. PHP uses its own memory, so the server workload and loading time is automatically reduced, which results in faster processing speed and better performance.

**Open Source:**

* + - * PHP source code and software are freely available on the web. You can develop all the versions of PHP according to your requirement without paying any cost. All its components are free to download and use.

**Familiarity with syntax:**

* + - * PHP has easily understandable syntax. Programmers are comfortable coding with it.

**Embedded:**

* + - * PHP code can be easily embedded within HTML tags and script.

**Platform Independent:**

* + - * PHP is available for WINDOWS, MAC, LINUX & UNIX operating system. A PHP application developed in one OS can be easily executed in other OS also.

**Database Support:**

* + - * PHP supports all the leading databases such as MySQL, SQLite, ODBC, etc.

**Error Reporting:**

* + - * PHP has predefined error reporting constants to generate an error notice or warning at runtime. E.g., E\_ERROR, E\_WARNING, E\_STRICT, E\_PARSE.

**Loosely Typed Language:**

* + - * PHP allows us to use a variable without declaring its datatype. It will be taken automatically at the time of execution based on the type of data it contains on its value.

**Web servers Support:**

* + - * PHP is compatible with almost all local servers used today like Apache, Netscape, Microsoft IIS, etc.

**Security:**

* + - * PHP is a secure language to develop the website. It consists of multiple layers of security to prevent threads and malicious attacks.

**Control:**

* + - * Different programming languages require long script or code, whereas PHP can do the same work in a few lines of code. It has maximum control over the websites like you can make changes easily whenever you want.

**A Helpful PHP Community:**

* + - * It has a large community of developers who regularly updates documentation, tutorials, online help, and FAQs. Learning PHP from the communities is one of the significant benefits.

**BACK-END**

**MySQL**

* + - * MySQL is an open-source relational database management system. Its 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 a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.

**What is Database?**

* + - * It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases available like MySQL, Sybase, Oracle, MongoDB, PostgreSQL, SQL Server, etc. In this section, we are going to focus on MySQL mainly.

**Features of MySQL**

**Relational Database Management System (RDBMS)**

* + - * MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.

**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.

**It is scalable**

* + - * MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, we can increase this number to a theoretical limit of 8 TB of data.

**Speed**

* + - * MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

**High Flexibility**

* + - * MySQL supports a large number of embedded applications, which makes MySQL very flexible.

**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).

**Allows roll-back**

* + - * MySQL allows transactions to be rolled back, commit, and crash recovery.

**Memory efficiency**

* + - * Its efficiency is high because it has a very low memory leakage problem.

**High Performance**

* + - * MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

**High Productivity**

* + - * MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.

**Platform Independent**

* + - * It can download, install, and execute on most of the available operating systems.

**Partitioning**

* + - * This feature improves the performance and provides fast management of the large database.

**GUI Support**

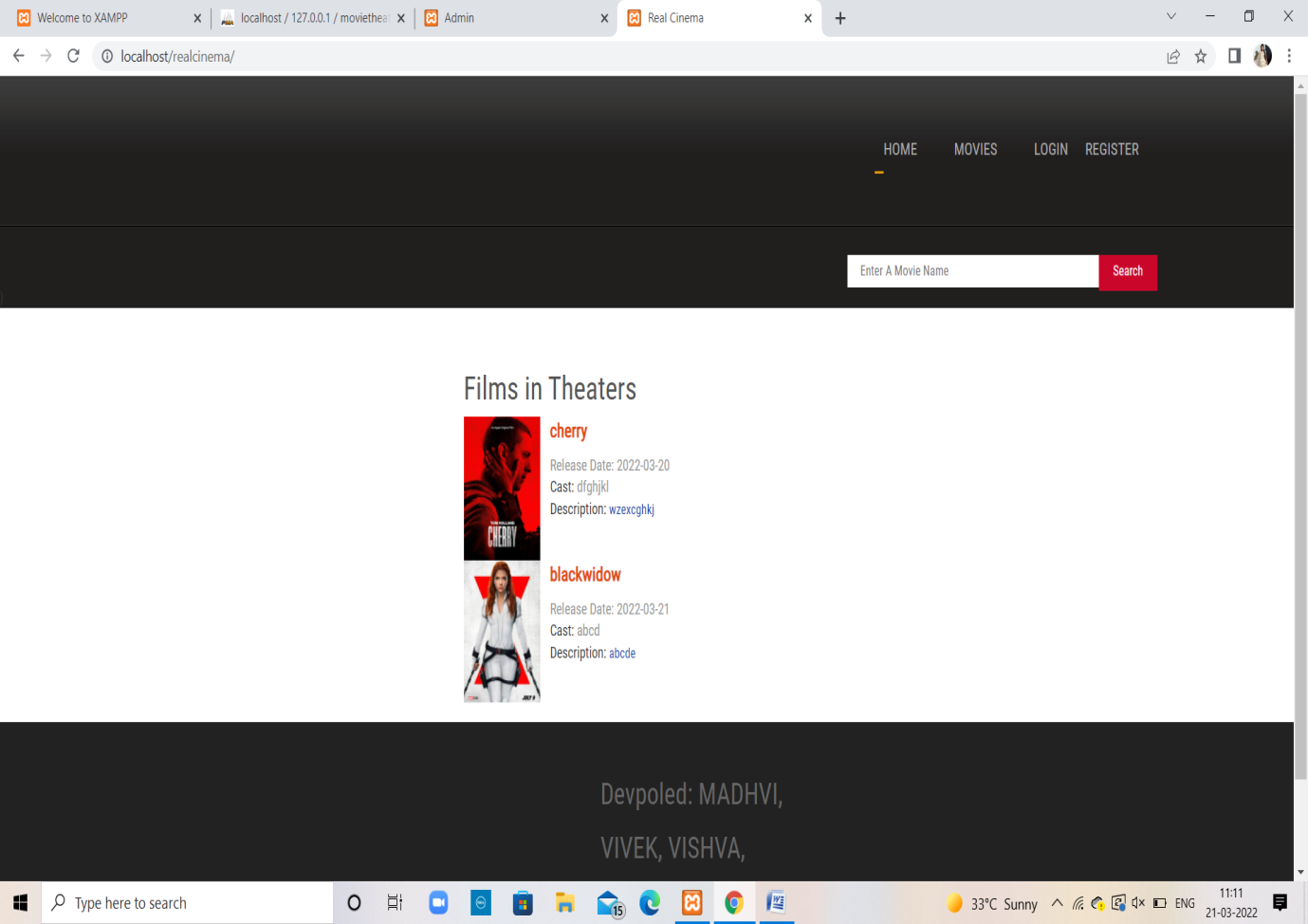
* + - * MySQL provides a unified visual database graphical user interface tool named "MySQL Workbench" to work with database architects, developers, and Database Administrators. MySQL Workbench provides SQL development, data modeling, data migration, and comprehensive administration tools for server configuration, user administration, backup, and many more. MySQL has a fully GUI supports from MySQL Server version 5.6 and higher.

**Dual Password Support**

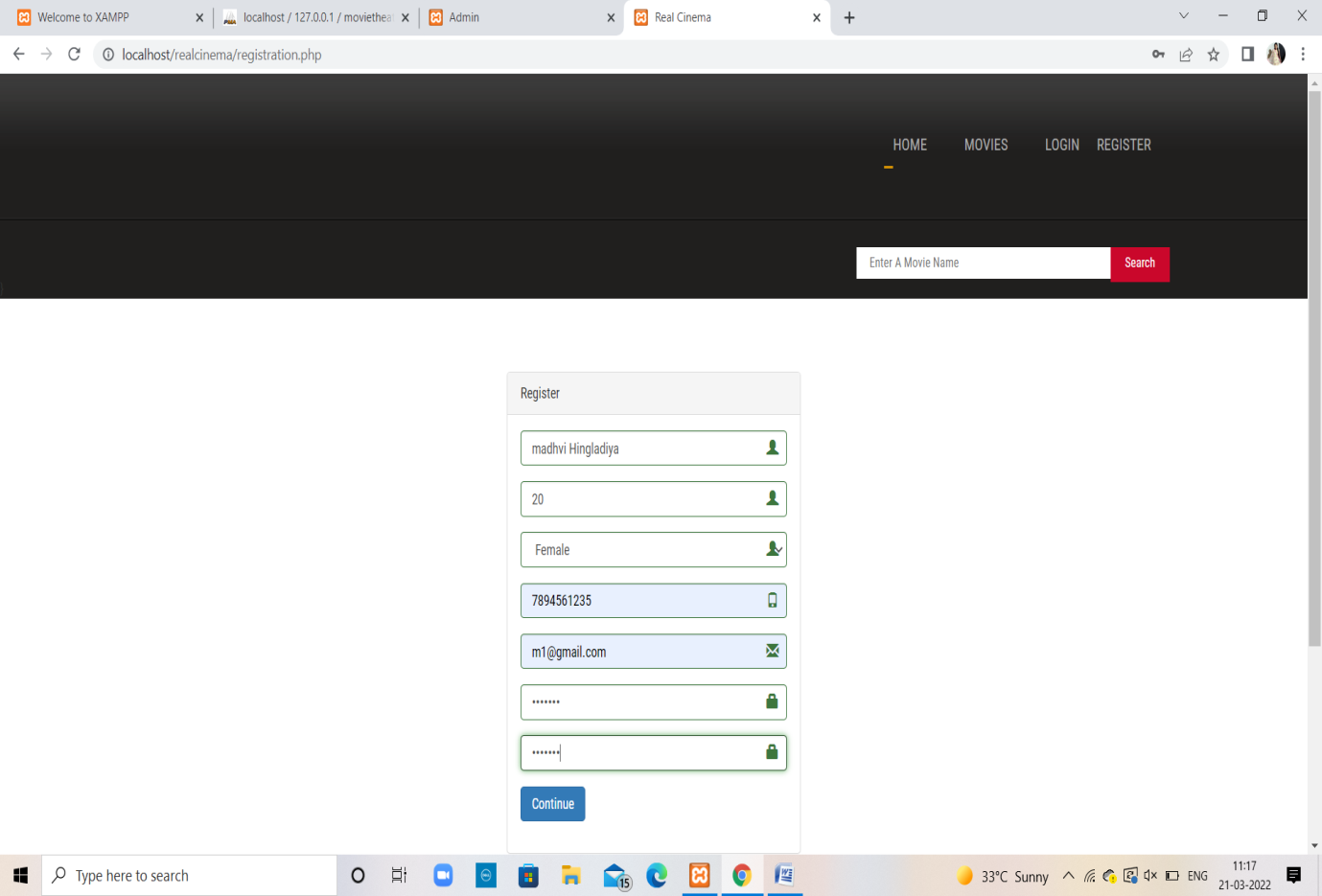
* + - * MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

**System design (Screen short after system implementation) :**

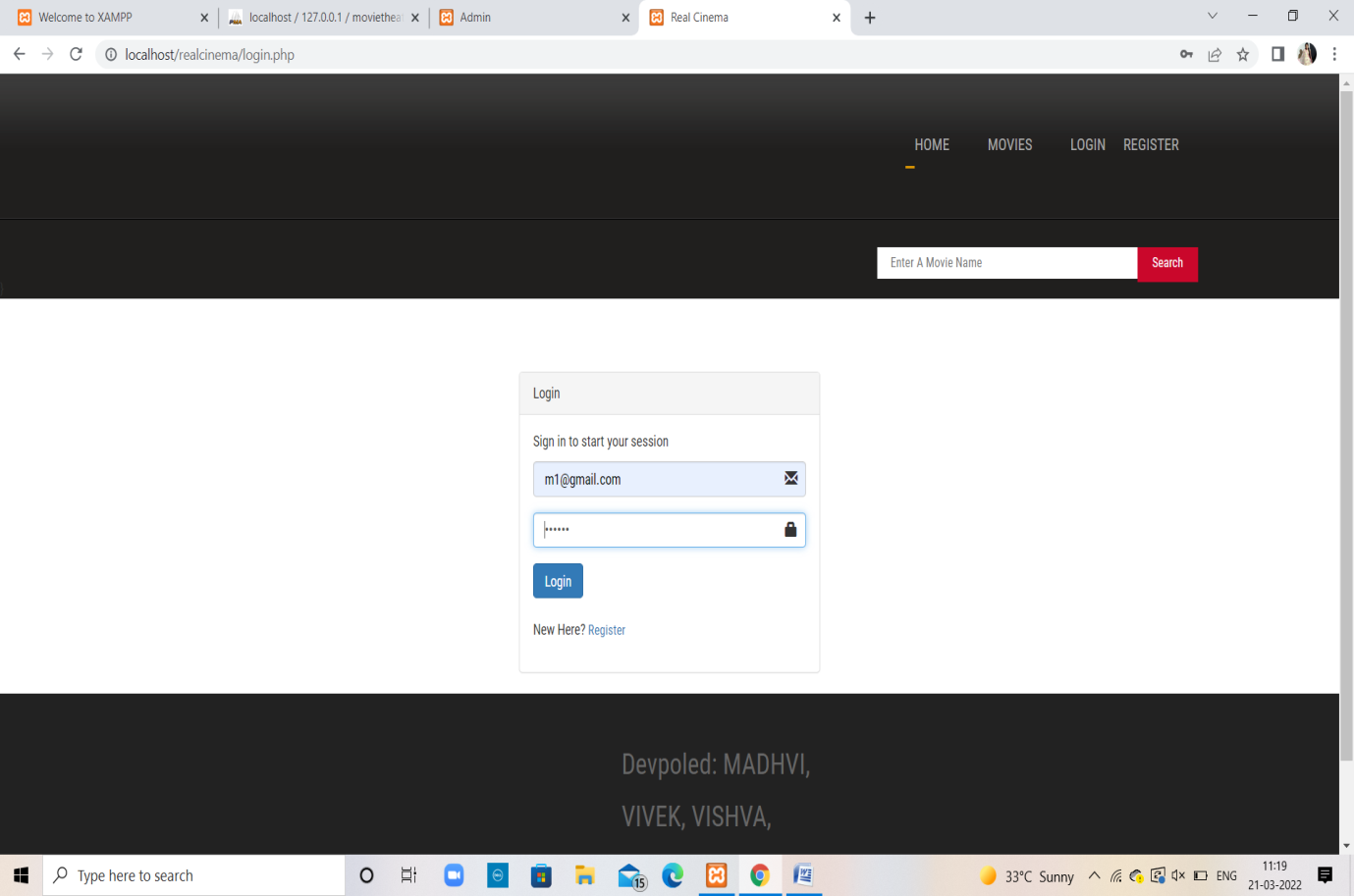
**Home page :**

****

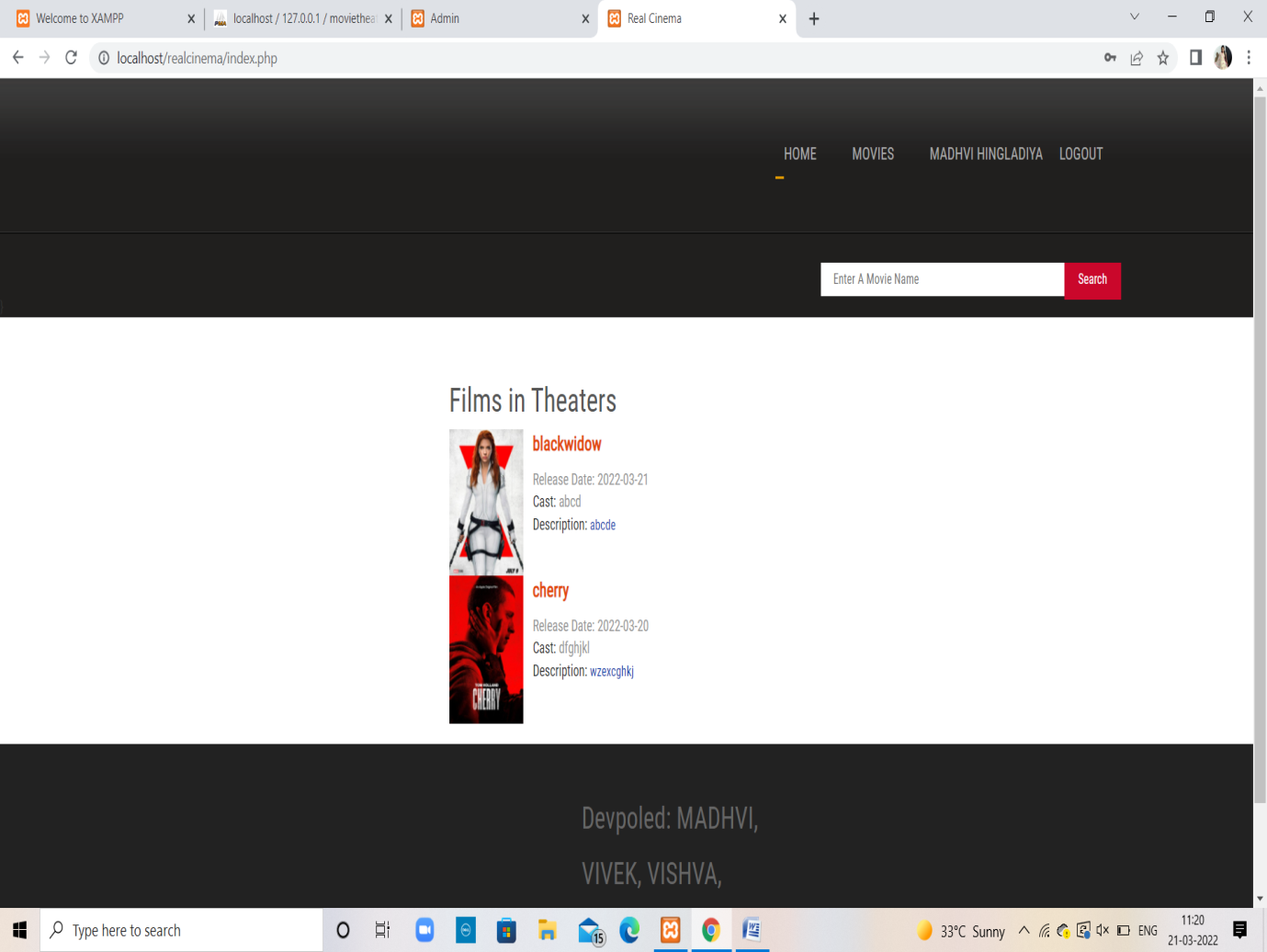
**Customer Registration page :**



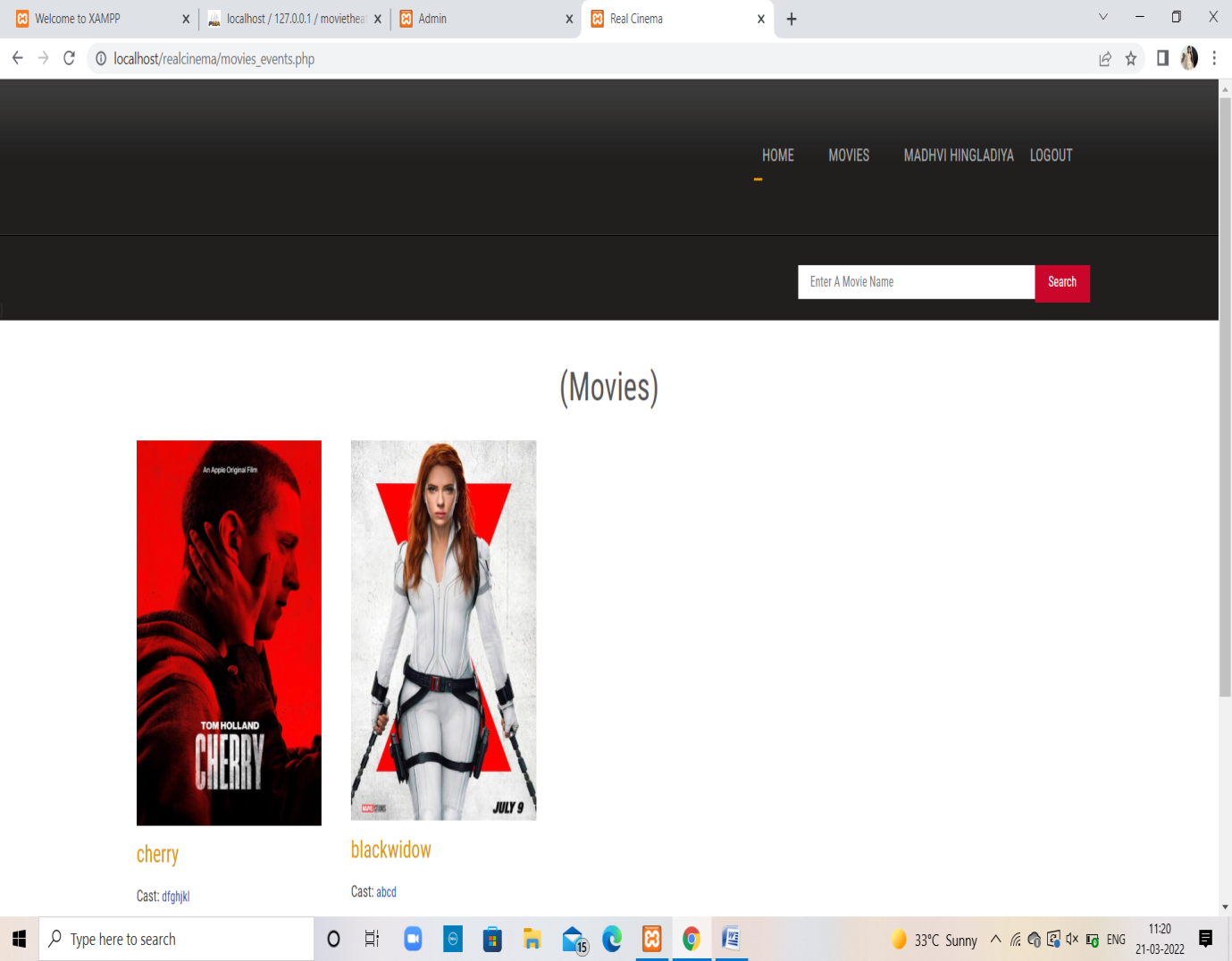
**Customer login page:**

****

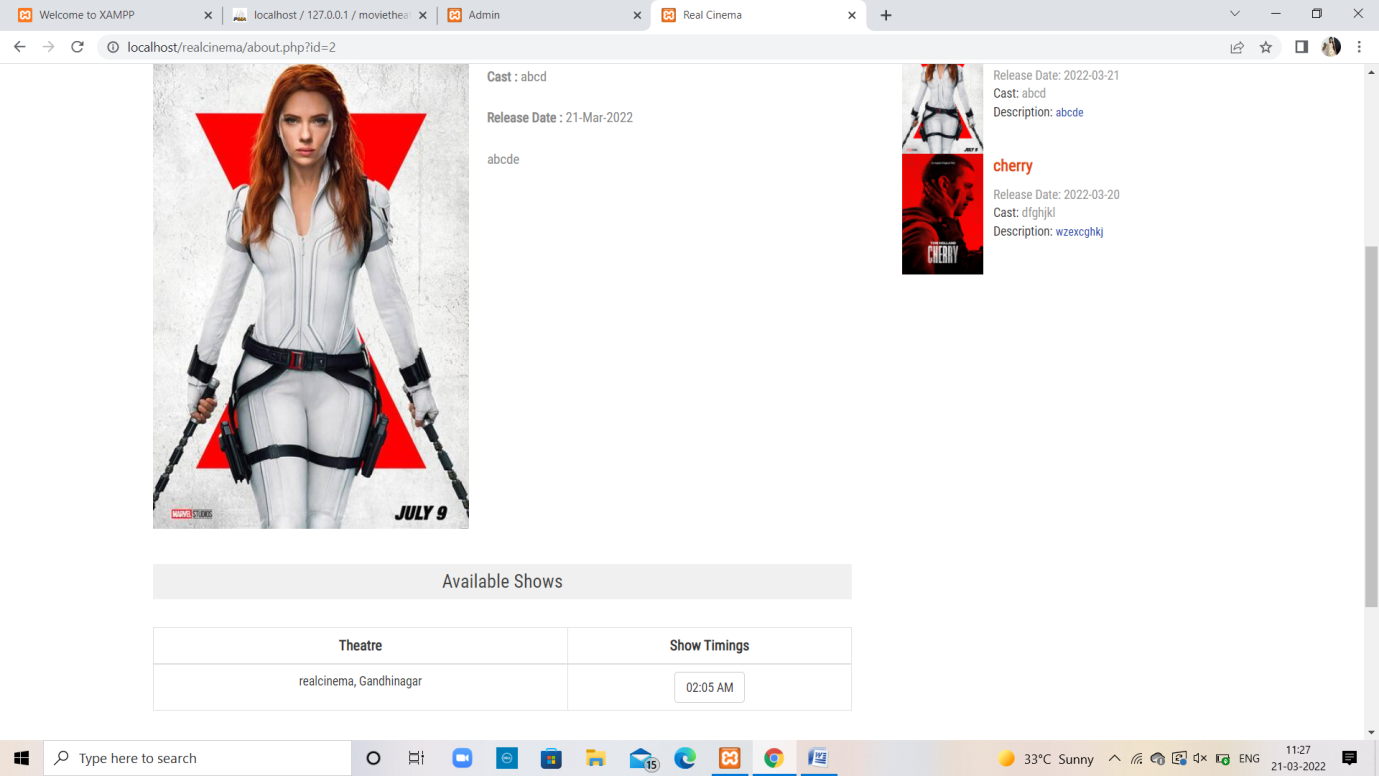
**Index page :**

****

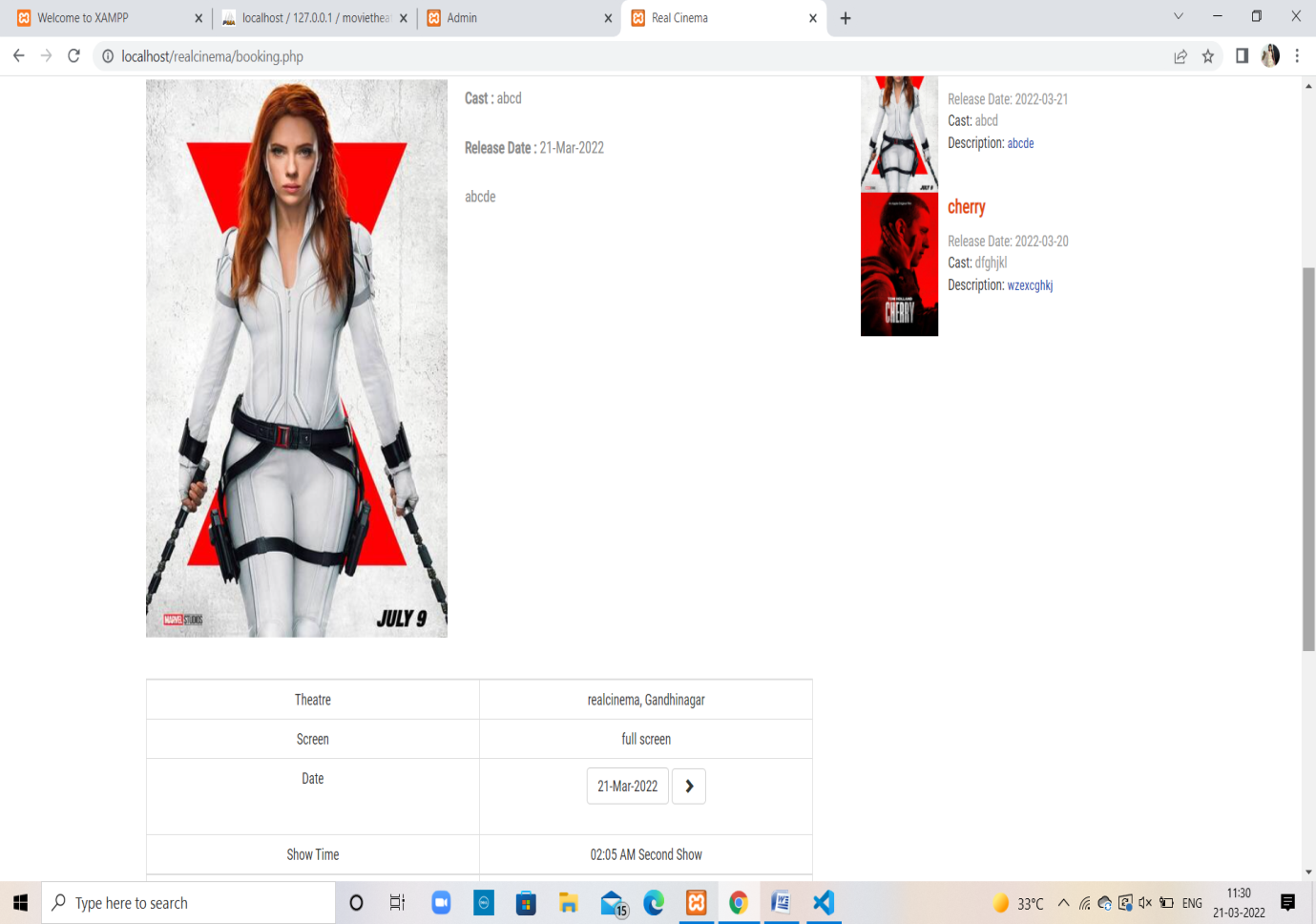
**Movie page :**

****

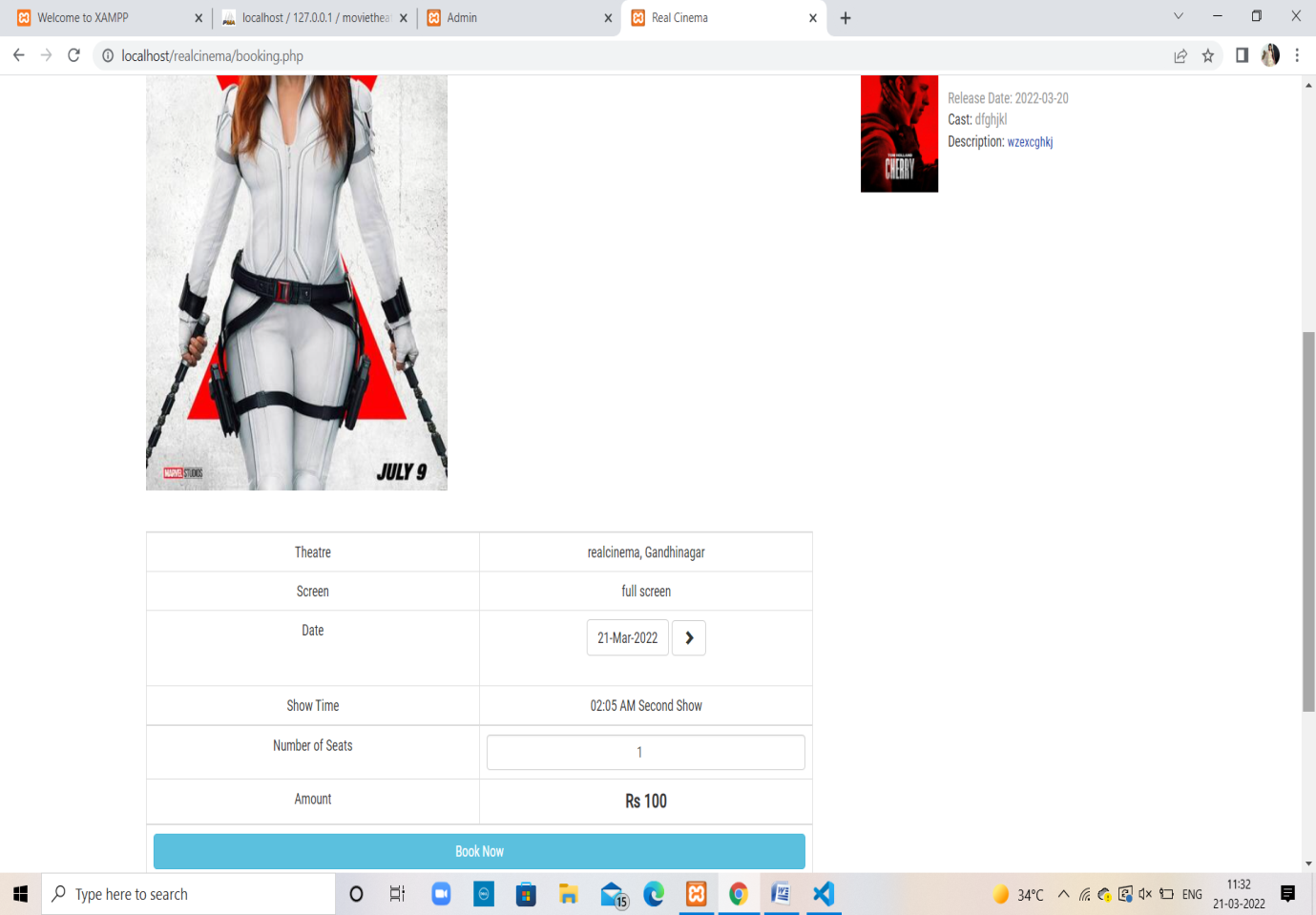
**Select movie :**

****

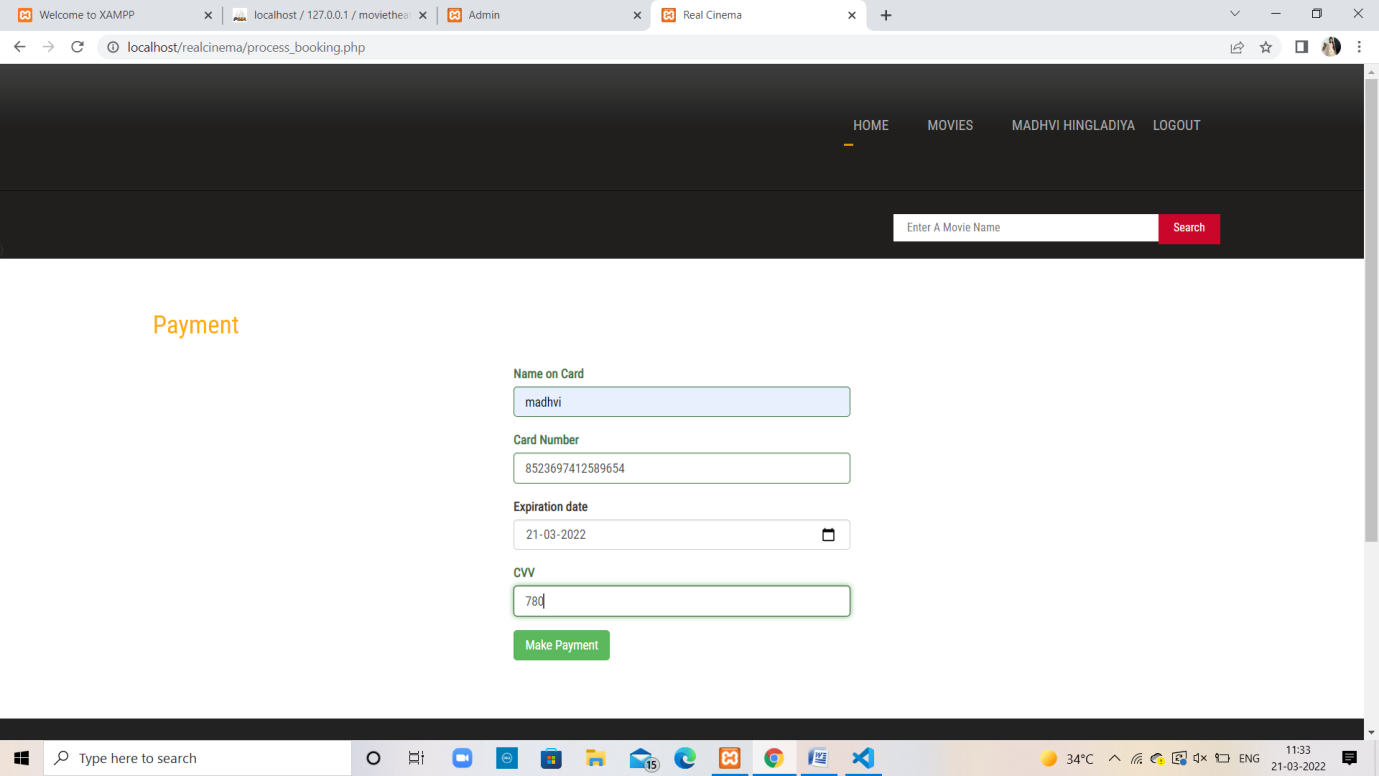
**Movies process page :**

****

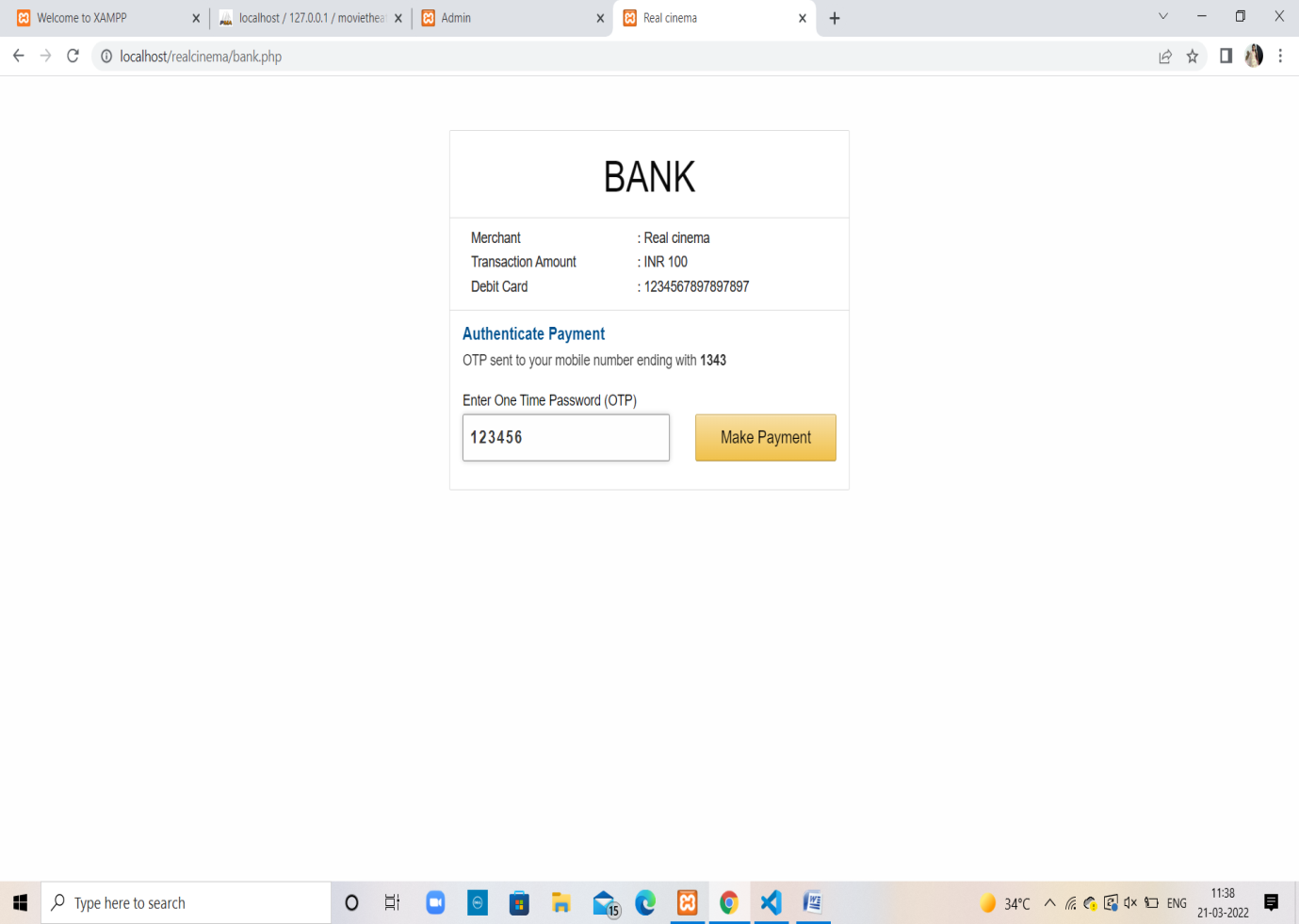
**Select seats peocess :**

****

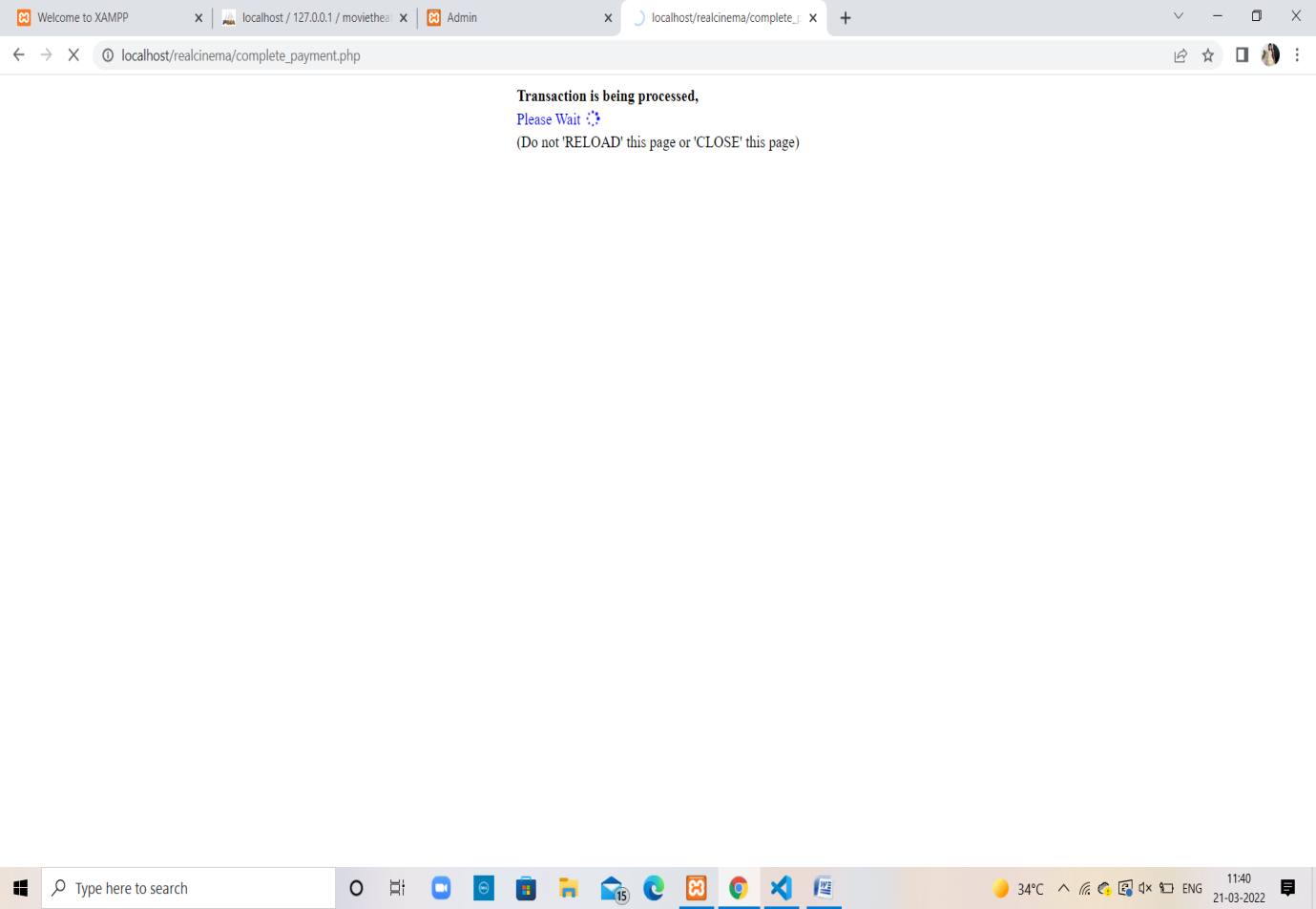
**Payment process :**

****

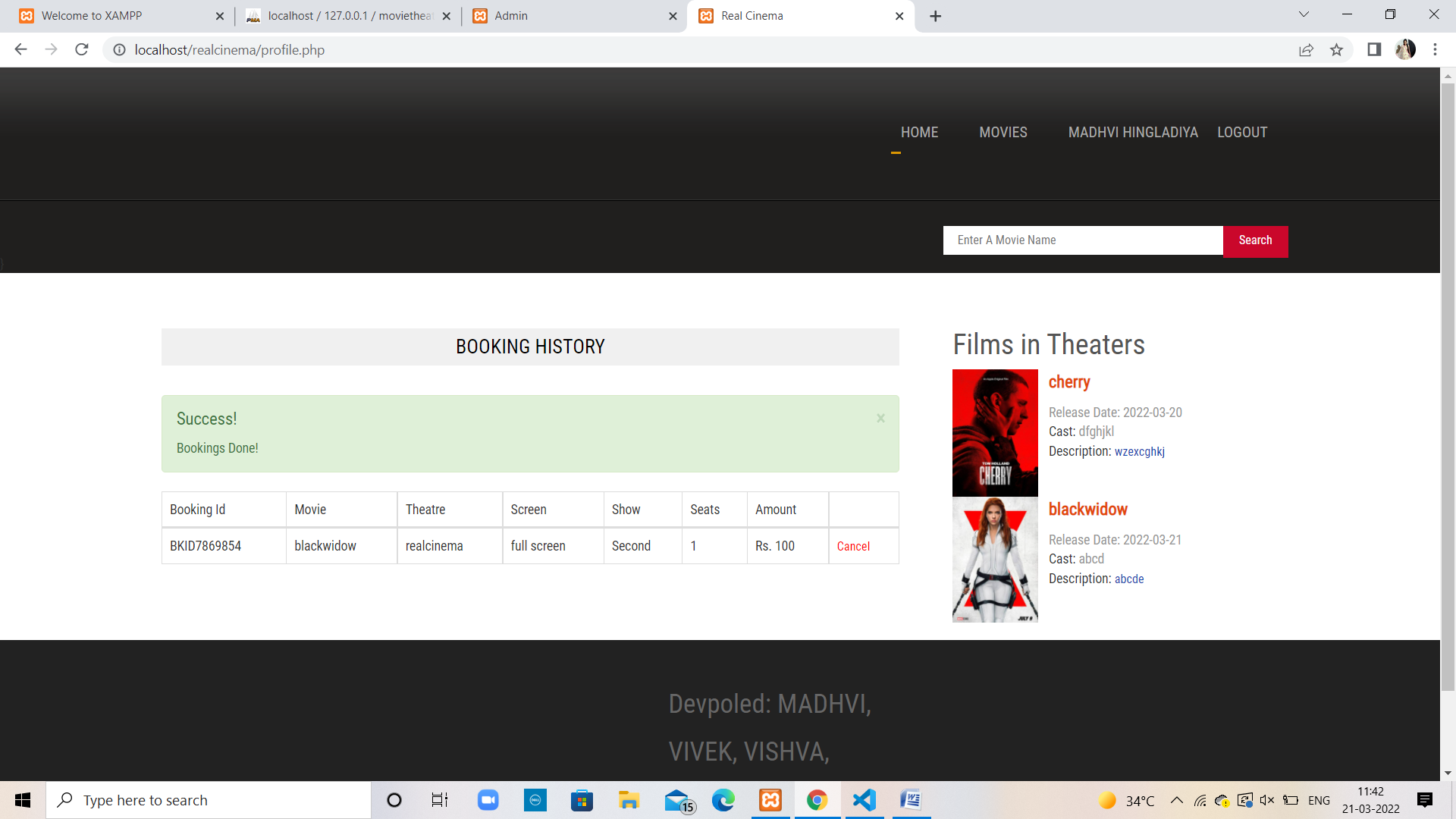
**OTP Process :**

****

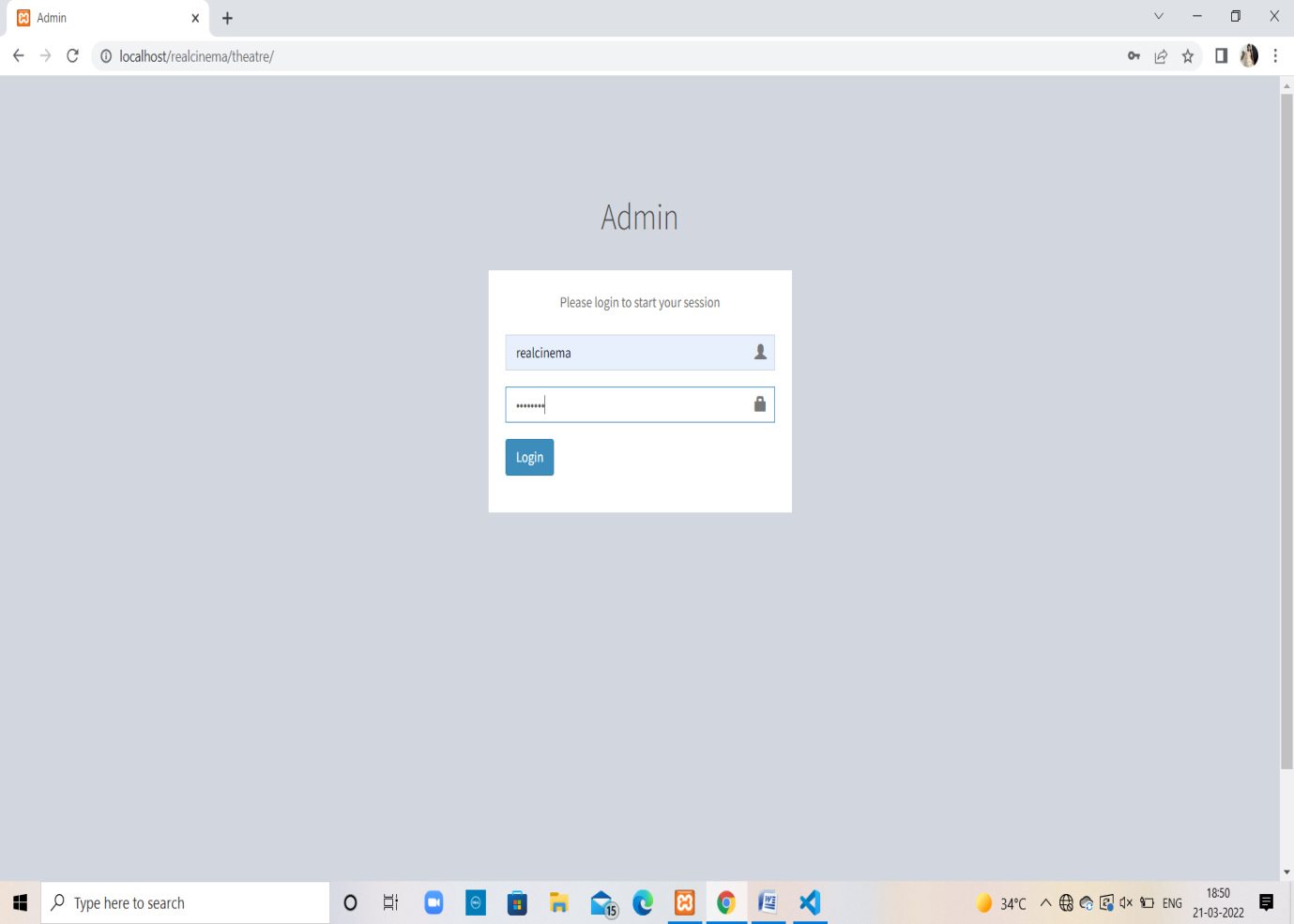
**Complete payment :**

****

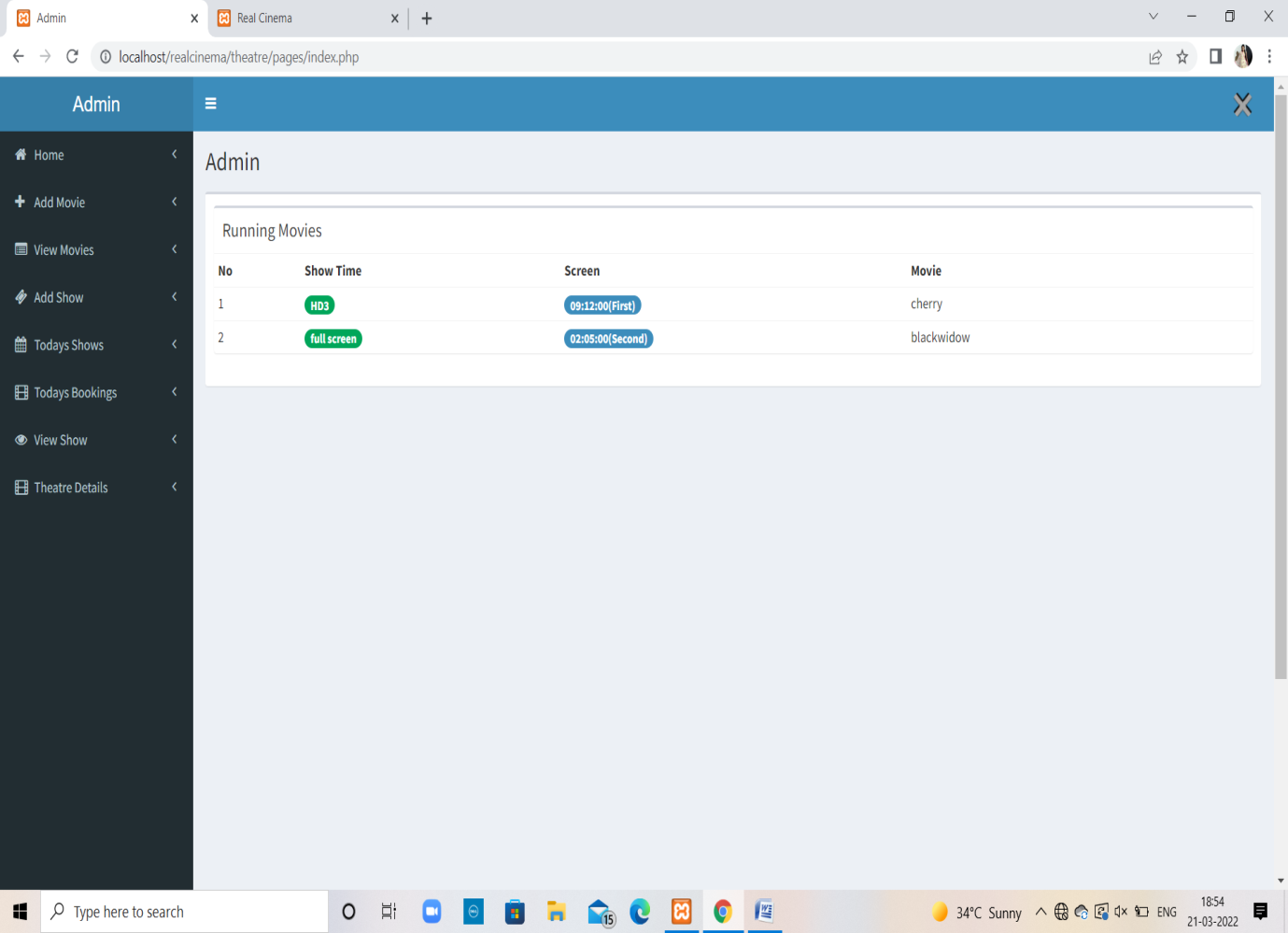
**Complete booking :**

****

**Admin Login :**



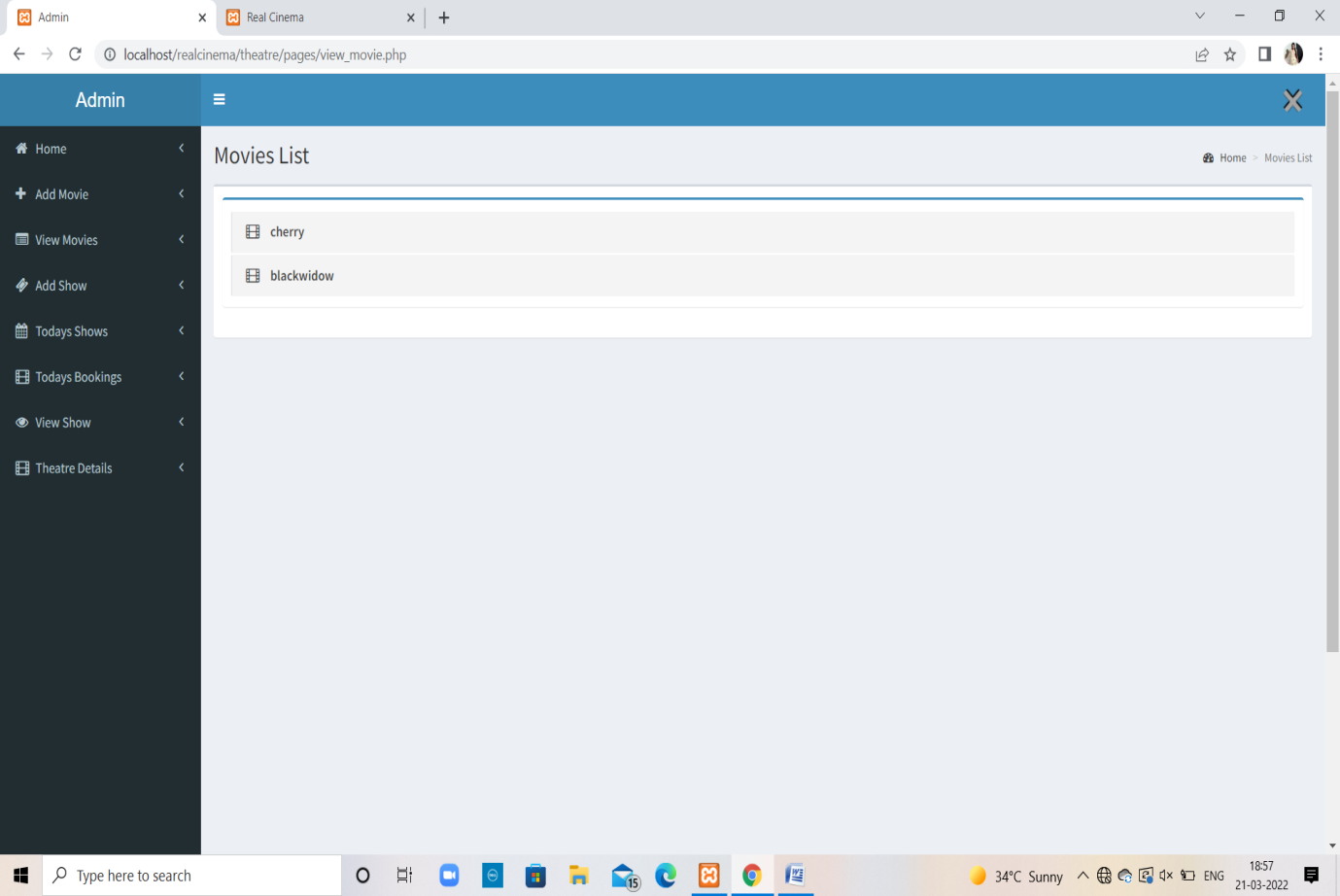
**Running Movies:**

****

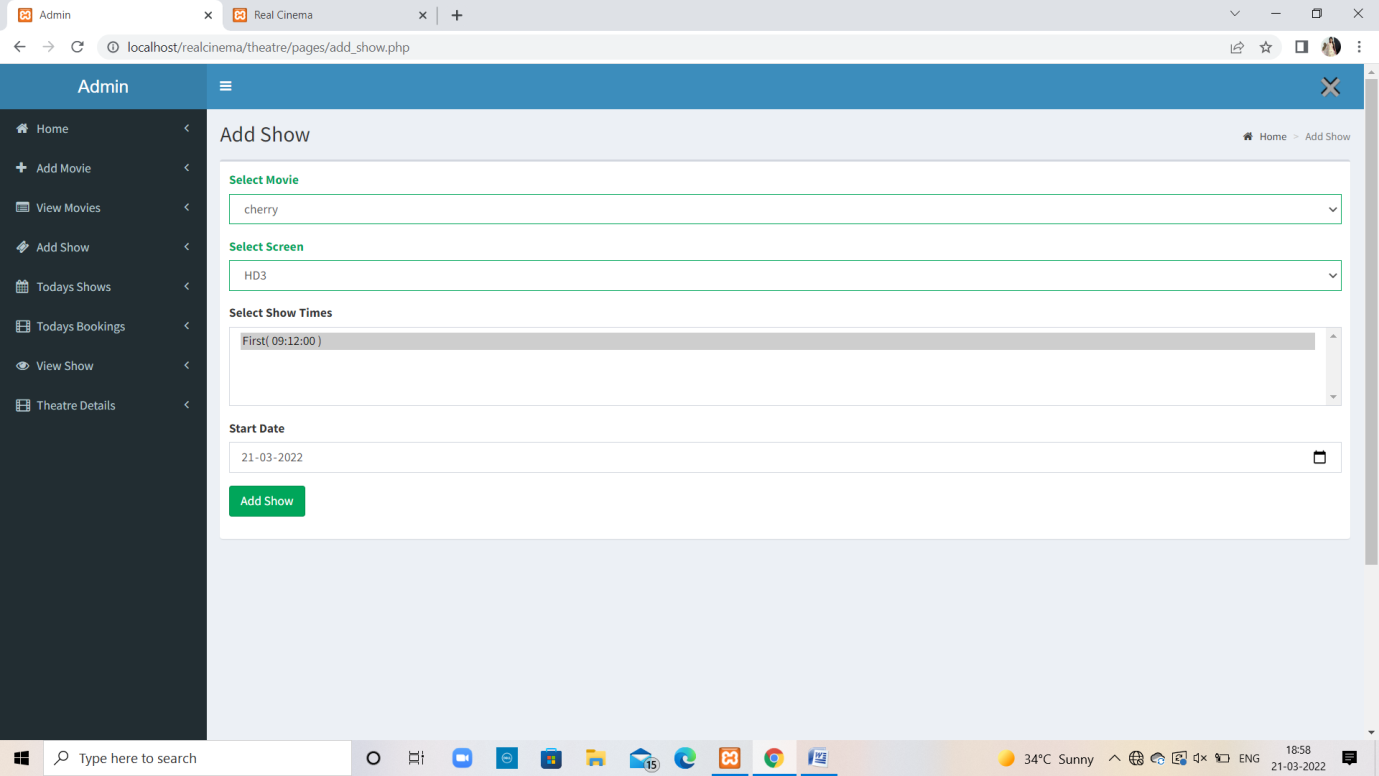
**Add Movie :**

****

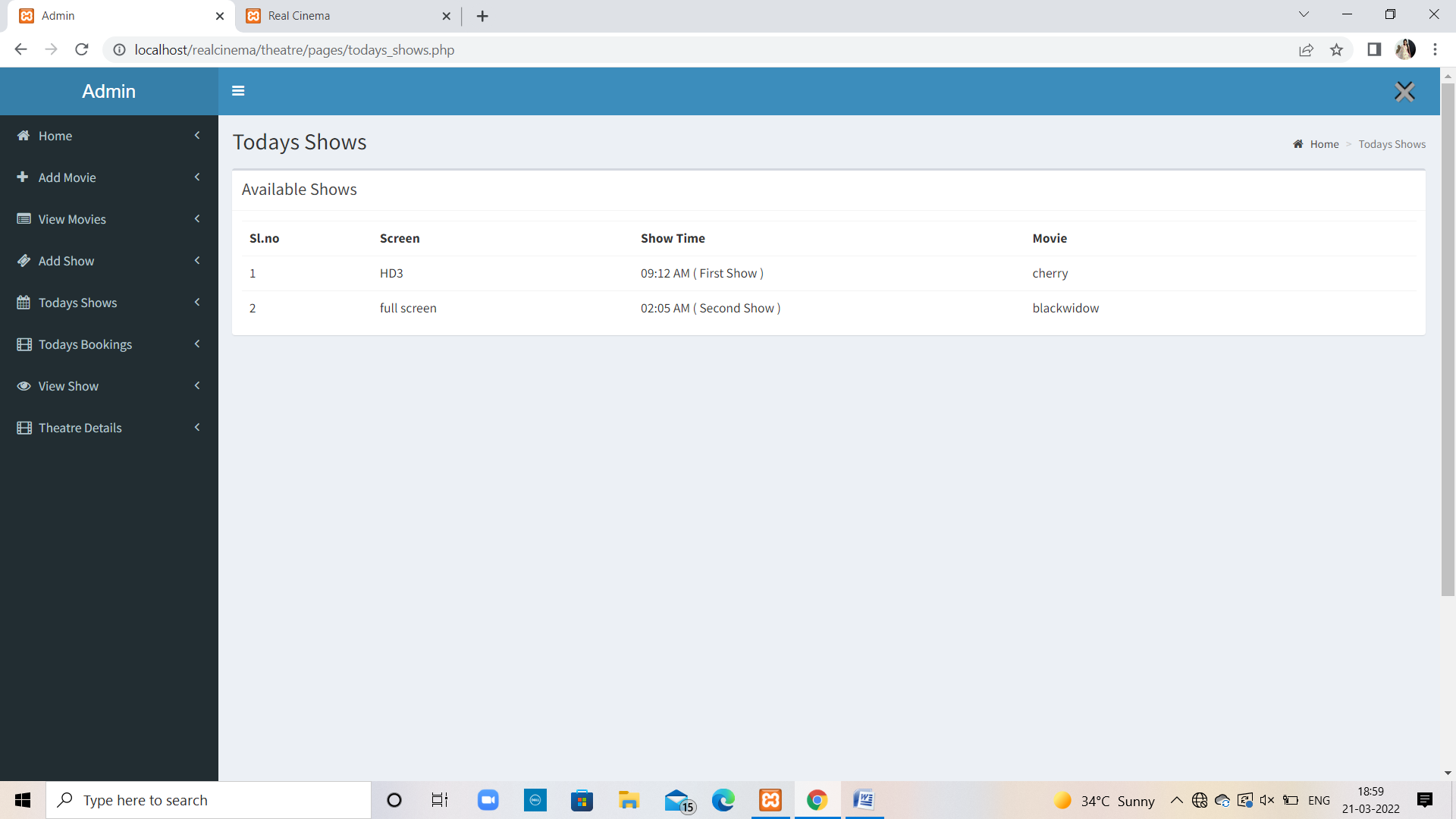
**Movies list :**

****

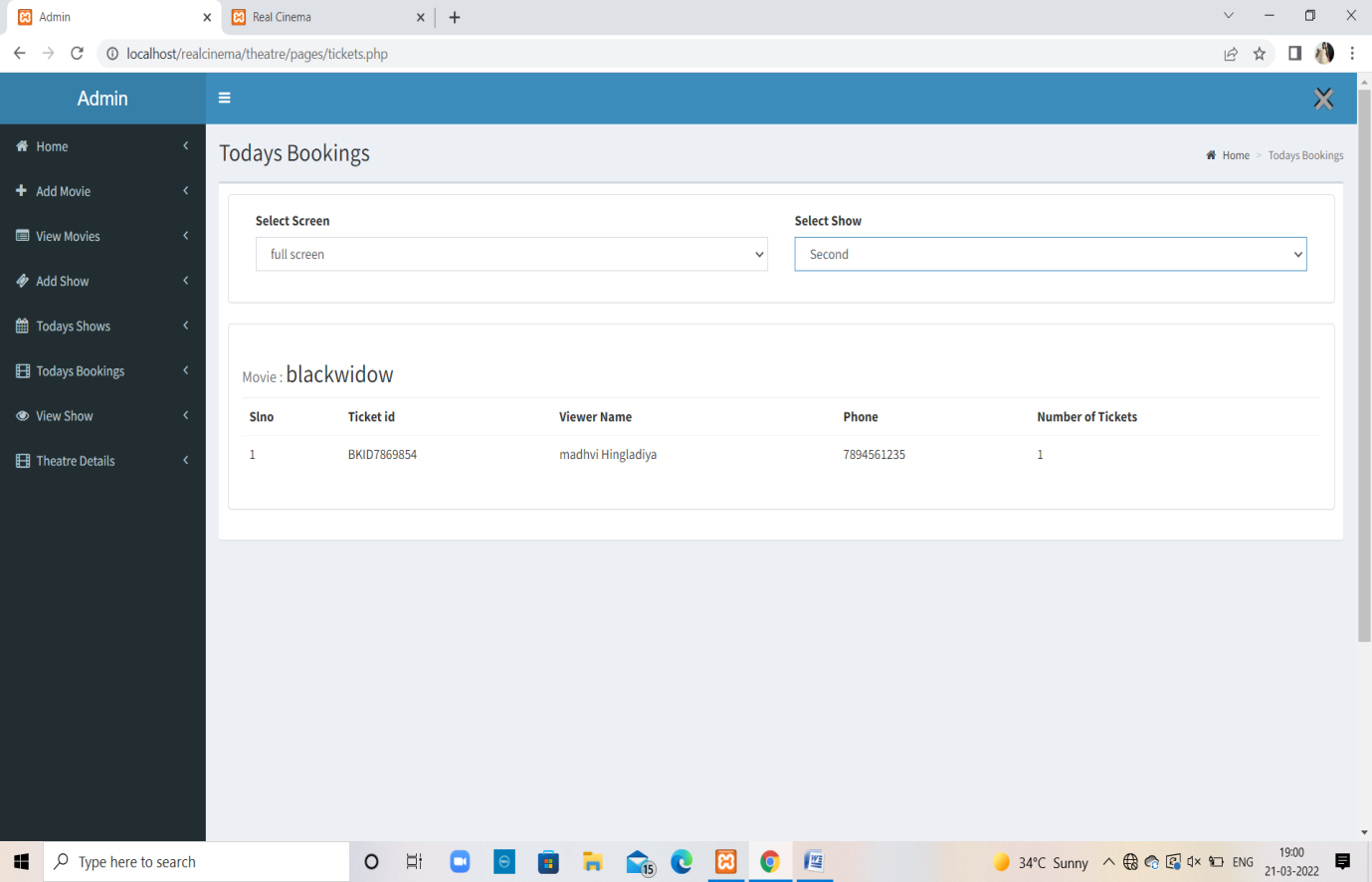
**Add show :**

****

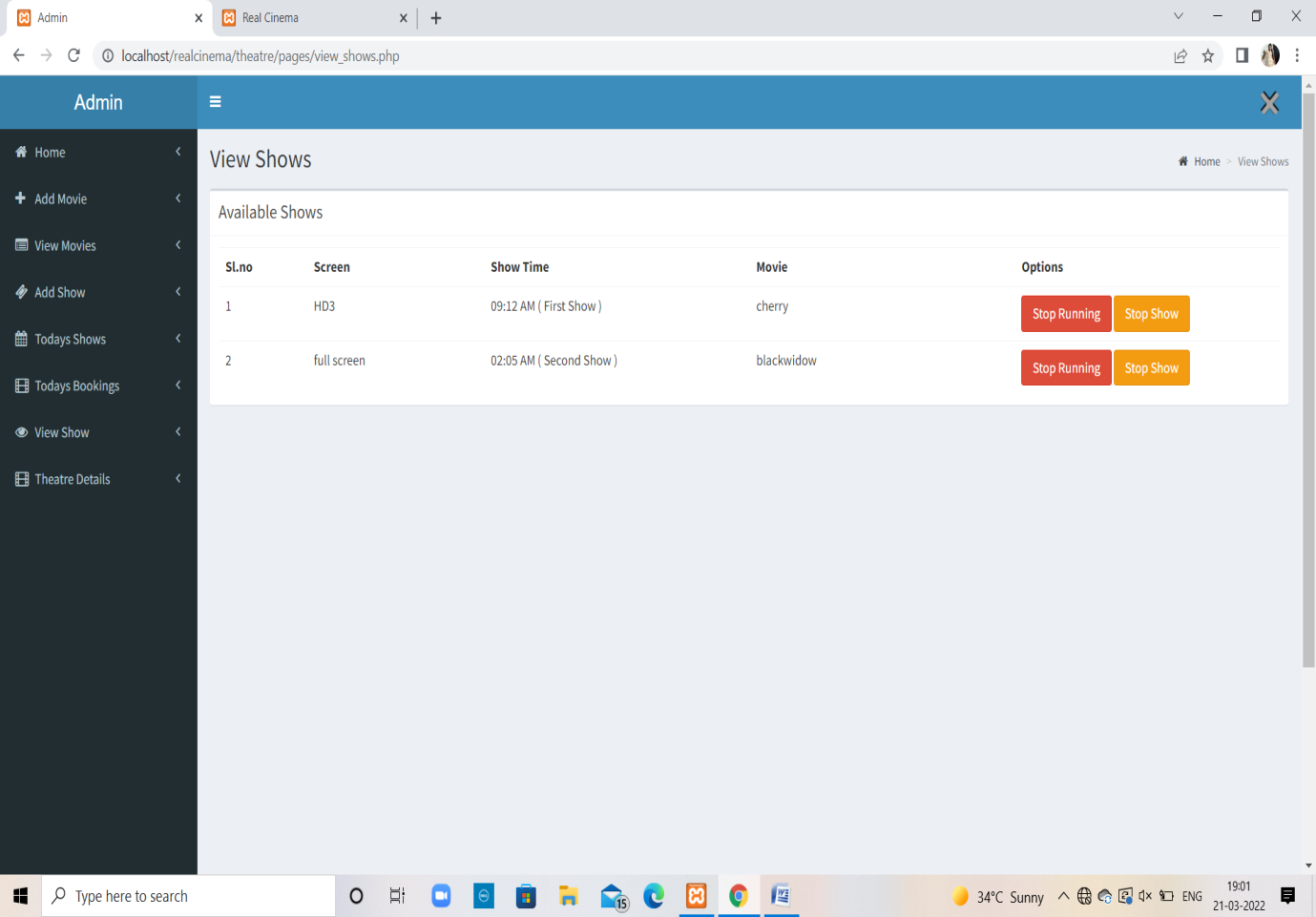
**Todays shows :**

****

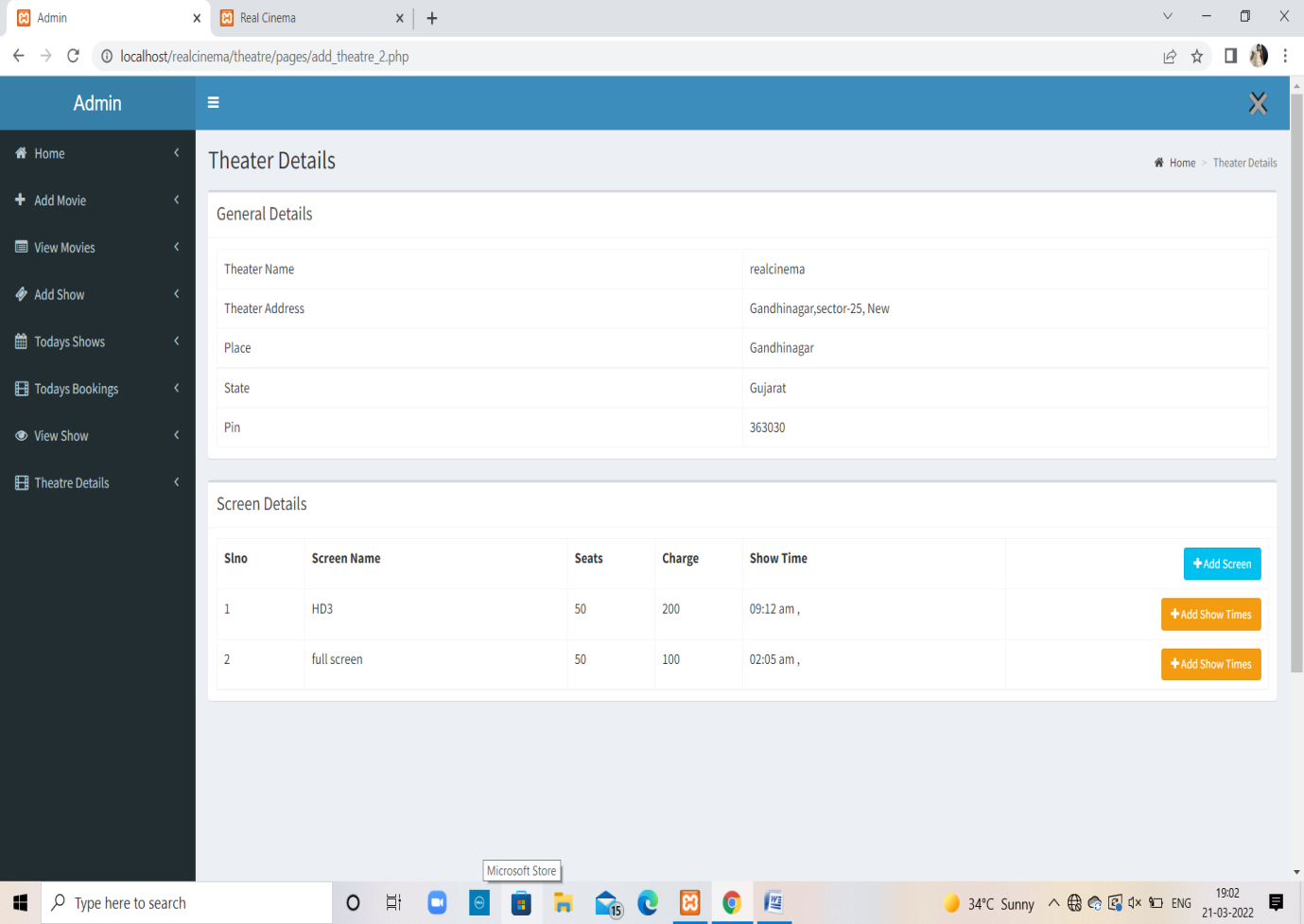
**Todays Bookings :**

****

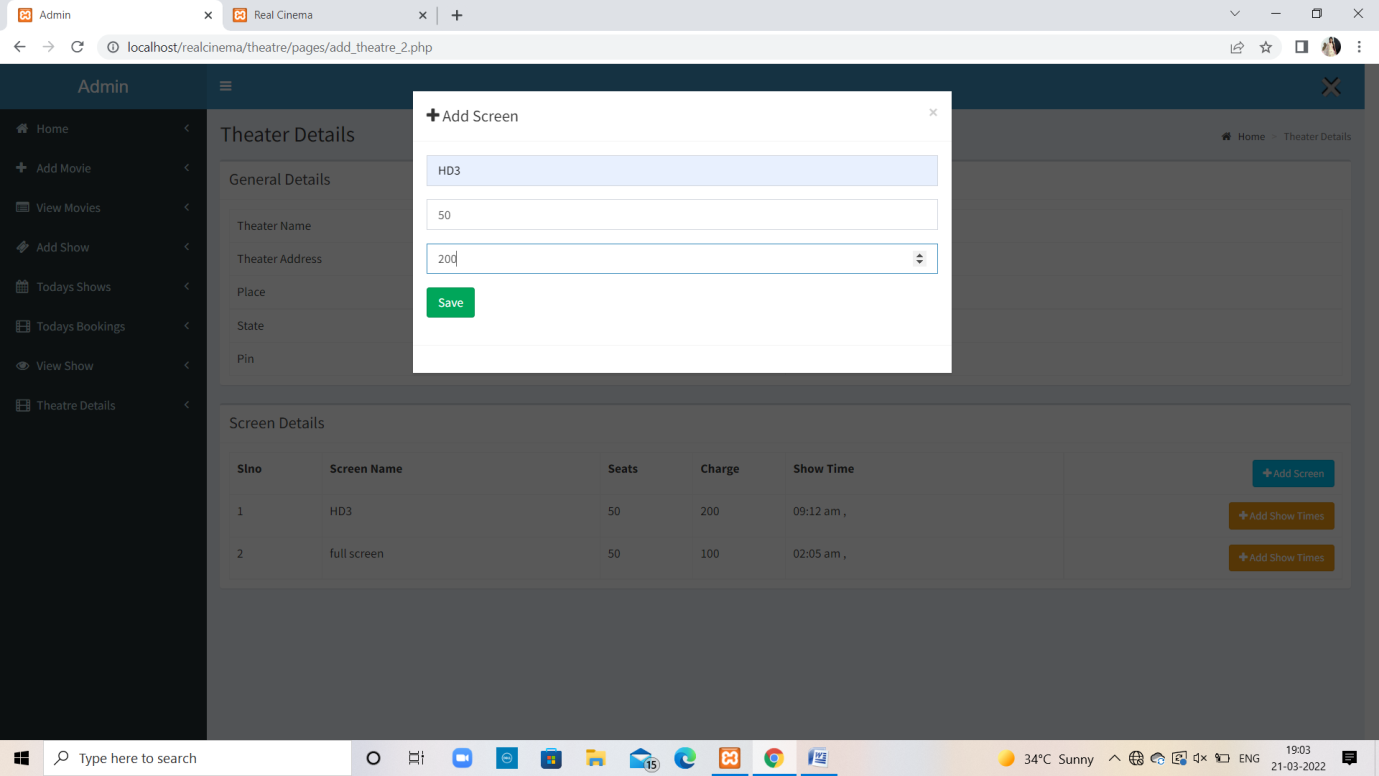
**View shows :**

****

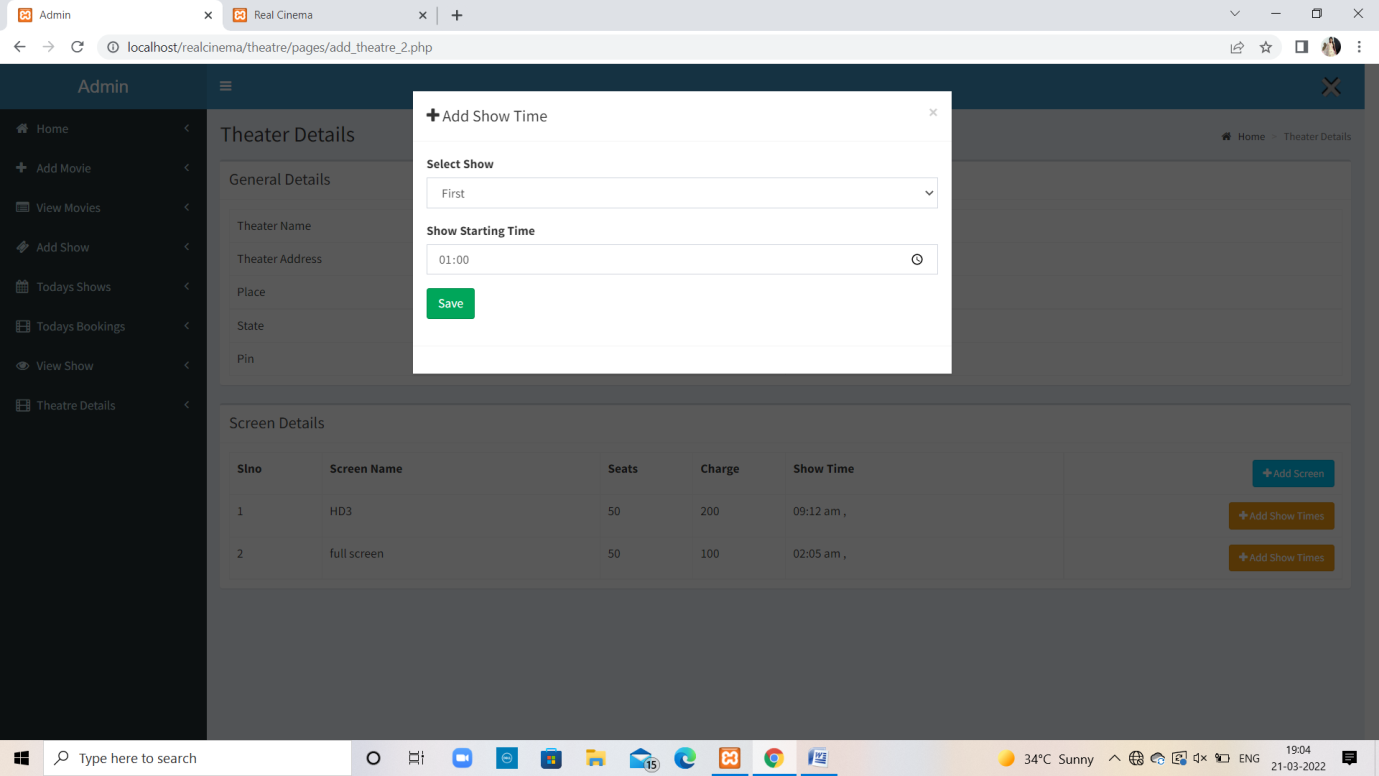
**Theater Details :**

****

**Add Screen :**

****

**Add Show Time :**

****