VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELGAUM-590014

A Mini-Project Report On

"Movie Booking System"

A Mini-project report submitted in partial fulfilment of the requirements for the award of the degree of ***Bachelor of Engineering in Computer Science and Engineering*** *of Visvesvaraya Technological University, Belgaum.*

Submitted by:

K RAJAT (1AM18CS076)

K SHREYAS (1AM18CS077)

PARTH H BHERWANI (1AM18CS124)

PAVAN SHREESHA B S (1AM18CS125)

Under the Guidance of:

Mrs. V.R. Srividhya

(Asst. Prof., Dept. of CSE)

Department of Computer Science and Engineering

AMC Engineering College,

18th K.M, Bannerghatta Main Road, Bangalore-560 083

2020-2021

AMC Engineering College,

18th K.M, Bannerghatta Main Road, Bangalore-560 083

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini-project work entitled **"MOVIE BOOKING SYSTEM"** has been successfully carried out by **K RAJAT (1AM18CS076), K SHREYAS (1AM18CS077), PARTH H BHERWANI (1AM18CS124), PAVAN SHREESHA B S (1AM18CS125)** bonafide students of **AMC Engineering College** in partial fulfilment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2020-2021. It is certified that all corrections/suggestions indicated for Internal Assessment 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 project work for the said degree.

Guide:

Mrs. SRIVIDHYA V.R.	**Dr. LATHA C.A.**	**Dr. A.G NATARAJ**
Asst.Prof., Dept. of CSE	**HOD, Dept. of CSE**	**Principal, AMCEC**

Examiners: **Signature with Date**

1.

2.

ACKNOWLEDGEMENT

The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. We are glad to express our gratitude towards our prestigious institution **AMC ENGINEERING COLLEGE** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express sincere thanks to our respected chairman **Dr. K. R. Paramahamsa** and beloved principal **Dr. A. G. Nataraj** for all their support.

We express our deepest gratitude and special thanks to **Dr. Latha C A**, **H.O.D, Dept. Of Computer Science And Engineering**, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project guide, **Mrs. V. R. Srividhya, Assistant Prof., Dept. Of Computer Science Engineering.**

K RAJAT (1AM18CS076)

K SHREYAS (1AM18CS077)

PARTH H BHERWANI (1AM18CS124)

PAVAN SHREESHA B S (1AM18CS125)

ABSTRACT

The entitled project "**MOVIE BOOKING SYSTEM**" is made keeping in mind all the aspects of the online websites. The project objective is to book cinema tickets online. The Ticket Reservation System is an Internet based application that can be accessed throughout the Net and can be accessed by anyone who has a net connection. This application will reserve the tickets. This online ticket reservation system provides a website for a cinema hall where any user of internet can access it. User is required to login to the system and needs to book the tickets. Tickets can be collected at the counter and watching movies with family and friends in theatres is one of the best medium of entertainment after having a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked. The website provides complete information regarding currently running movies on all the screens with details of show timings, available seats, snacks and movie details.

To develop this project we need basics of HTML , CSS , JS , SQL and PHP . We have made the most friendly and easy access user interface for our users. We have made use of HTML , CSS , JS for our front end development . The backend is made use of MYSQL , Maria DB and PHP . Hosting is done locally in our own system using a XAMPP Server which contains the Mysql database .

INDEX

1. INTRODUCTION..... 6

1.1 Introduction to DBMS

1.2 Introduction to SQL

1.3 Introduction to Movie Booking System

1.4 Scope and Importance of work

2. SOFTWARE AND HARDWARE REQUIREMENT..... 8

2.1 Software Requirements

2.2 Hardware Requirements

3. DESIGN..... 9

3.1 ER Diagram

3.2 Schema Diagram

3.3 Normalization of tables

4. IMPLEMENTATION..... 13

4.1 Front-End code

4.2 Back-End code

4.3 DB access and connectivity

4.4 Connection code

4.5 Triggers

4.6 Stored Procedures

5. SNAPSHOTS..... 30

6. CONCLUSION..... 37

6. REFERENCES..... 38

INTRODUCTION

1.1 Introduction to DBMS

A database is simply an organized collection of related data, typically stored on disk, and accessible by possibly many concurrent users. Databases are generally separated into application areas. For example, one database may contain Human Resource (employee and payroll) data; another may contain sales data; another may contain accounting data; and soon. Databases are managed by a DBMS. The choice of a database product is often influenced by factors such as: the computing platform (i.e., hardware, operating system)

- the volume of data to be managed
- the number of transactions required per second
- existing applications or interfaces that an organization may have
- support for heterogeneous and/or distributed computing
- cost
- vendor support

1.2 Introduction to SQL

SQL, which is an abbreviation for Structured Query Language, is a language to request data from a database, to add, update, or remove data within a database, or to manipulate the metadata of the database.

SQL is a declarative language in which the expected result or operation is given without the specific details about how to accomplish the task. The steps required to execute SQL statements are handled transparently by the SQL database. Sometimes SQL is characterized as non-procedural because procedural languages generally require the details of the operations to be specified, such as opening and closing tables, loading and searching indexes, or flushing buffers and writing data to filesystems. Therefore, SQL is considered to be designed at a higher conceptual level of operation than procedural languages because the lower level logical and physical operations aren't specified and are determined by the SQL engine or server process that executes it.

1.3 Introduction to Movie Booking System

Online Movie ticket booking system is the process by which customers can purchase their movie tickets directly using internet and pay through online banking. It is useful for both customers and promoters equally. Customers can make online ticket booking at any instance of 24 hours a day and as this is web based application they can buy ticket from anywhere around the world.

Customers with cinema website to know about new movies, timings and cinema location. Customers can buy ticket without wasting their time. Cinema owner can inform the customer about the new release and also can get feedback from customer as it will be helpful for him to make new strategy for business.

Applications of the project

• Online movie booking project is aimed to provide facility to book cinema ticket from anywhere at anytime.

• One objective of this project is to minimize the number of staff at ticket-box window.

• Provide a 24*7 service to the customer.

Up till now all the movie ticket booking are performed manually it requires more human efforts to manage each and every aspects of organization which is very costly process. To overcome all kind of problems faced by movie ticket booking process our project “Movie Ticket Booking Management” will provide efficient and cost effective solution. The project “Movie Ticket Booking Management” is generalized software and can be easily used by any multiplex with little or no change. The changes in software can easily be accommodated. The addition and deletion of the modules in software can be easily adjusted.

1.4 Scope and importance of work

The scope of the project is clear to give a simple and attractive application to simplify the work as well as to reduce the efforts while doing it offline. In this application we are able to save the database of all the admin, user, movie data and ticket details. In this mini project we are able to save the data of all user ID, movie data and ticket details.

SOFTWARE AND HARDWARE REQUIREMENTS

A computerized way of handling information about property and users' details is efficient, organized and time saving, compared to a manual way of doing so. This is done through a database driven web application whose requirements are mentioned in this section.

2.1 SOFTWARE REQUIREMENTS:

Software used:

- Operating System – Linux
- Front End – Visual Studio Code
- Back End – LAMPP/ phpmyadmin

Technology used:

- Front End – HTML/ CSS/ JS
- Back End – MySQL

2.2 HARDWARE REQUIREMENTS:

Hardware Components used:

- CPU: Intel Core i3
- RAM: 4 GB
- Peripherals: USB Keyboard, Optical Mouse

DESIGN

3.1 ER DIAGRAM:

An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business.

An E-R model does not define the business processes; it only presents a business data schema in graphical form. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities.

Entities may be characterized not only by relationships, but also by additional properties (attributes), which include identifiers called "primary keys". Diagrams created to represent attributes as well as entities and relationships may be called entity-attribute-relationship diagrams, rather than entity-relationship models.

An ER model is typically implemented as a database. In a simple relational database implementation, each row of a table represents one instance of an entity type, and each field in a table represents an attribute type. In a relational database a relationship between entities is implemented by storing the primary key of one entity as a pointer or "foreign key" in the table of another entity

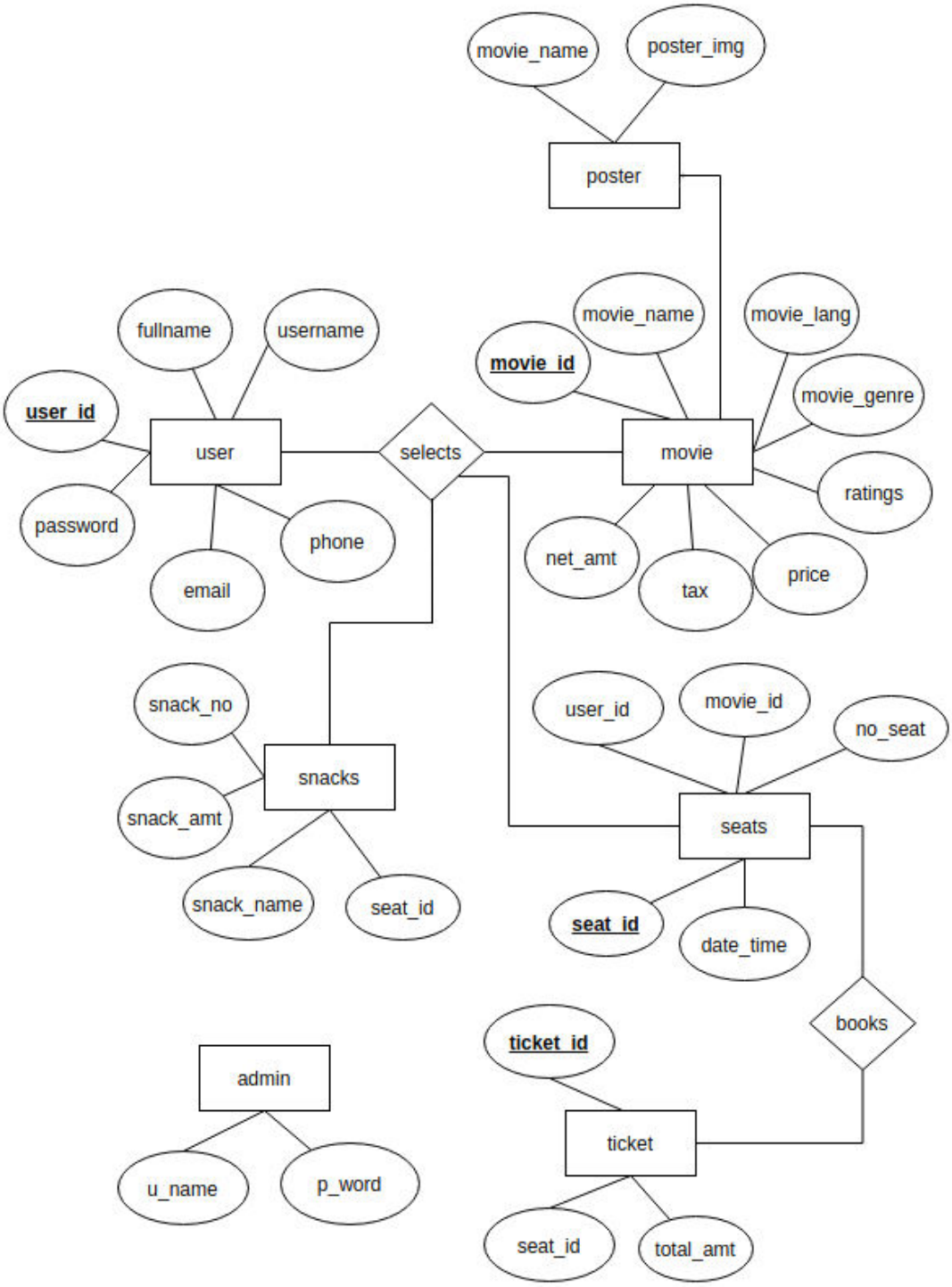

Fig 3.1: ER Diagram

3.2 SCHEMA DIAGRAM

The design of the database is called a schema. This tells us about the structural view of the database. It gives us an overall description of the database. A database schema defines how the data is organised using the schema diagram. A schema diagram is a diagram which contains entities and the attributes that will define that schema. A schema diagram only shows us the database design. It does not show the actual data of the database. Schema can be a single table or it can have more than one table which is related. The schema represents the relationship between these tables.

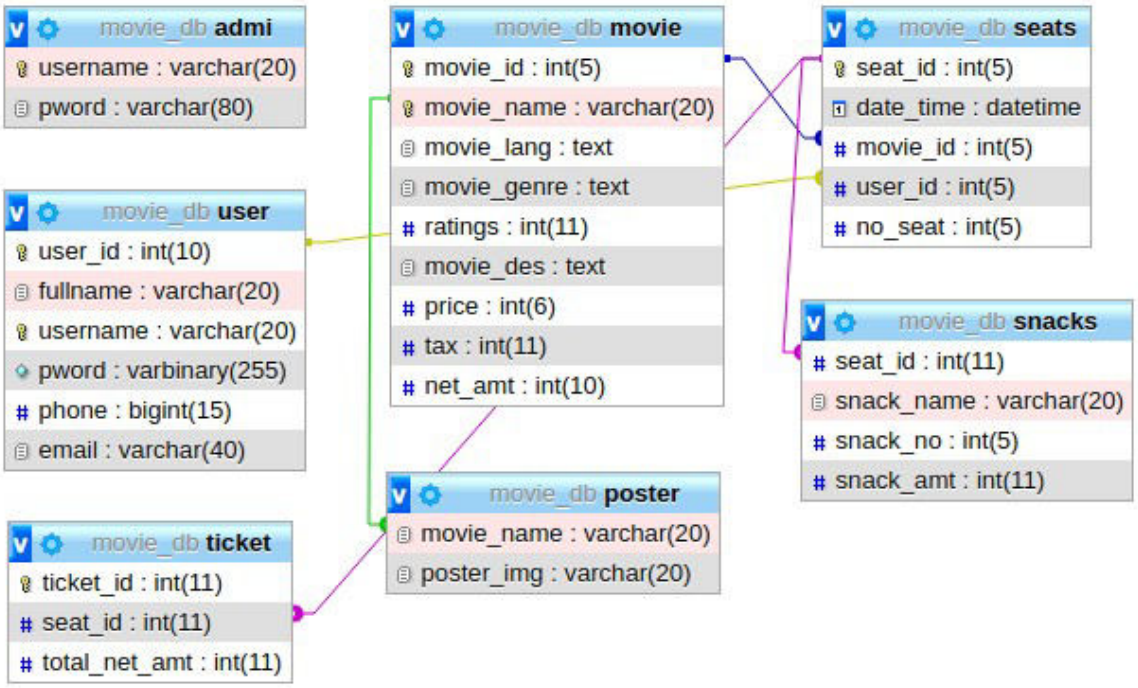

Fig 3.2: Schema Diagram

3.3 Normalizations of Tables :

The normalization process, as first proposed by Codd (1972), takes a relation schema as Series of tests to clarify whether it satisfies a certain normal norm.
The process which proceeds in a top down fashion by evaluating the each relation against the Criteria for normal forms and decomposing relations can be necessary, can thus be considered as relational design by analysis.

Normalization of data can be considered as a process of analyzing the given relation schemas based on their FDs and primary keys to achieve the desirable properties of (1) minimizing redundancy and (2) minimizing insertion, deletion and update anomalies.

First normal form (1NF)

It states that the domain of an attribute must include only atomic (simple, indivisible) Values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

Hence, 1NF disallows having a set of values, tuple of values, or a combination of both as an attribute value for a single tuple. In other words 1NF disallows relations within relations or relations as an attribute values within tuples.

Second normal form (2NF)

A relation schema R is in 2NF if every nonprime attribute A in R is fully functionally dependent on the primary key of R.

Third normal form (3NF)

A relation schema R is in third normal form (3NF) if, whenever a nontrivial functional dependency X->A holds in R, either (a) X is a super key of R, or (b) A is a prime attribute of R.

Here, all the tables depict Normalization upto 3rd normal form.

IMPLEMENTATION

4.1 FRONT END CODE:

Technology used: HTML, CSS, JavaScript

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

A. User Registration

```
<!DOCTYPE html>
<html>
<link rel="stylesheet" href="css/styles.css">
  <head>
    <title>Movie and Chill</title>
  </head>
  <body>
  <div class="user_login">
  <form action="UserRegQ.php" method="POST" class="inputform">
  <img src="assets/icons/laptop-user.png"><br></br>
  <input type="text" id="fullname" name="fullname" placeholder=" Fullname" required/>
  <input type="text" id="username" name="username" placeholder=" Username"
required/><br><br>
  <input type="password" id="password" name="password" placeholder=" **********"
required/>
  <input type="password" id="password" placeholder=" Confirm Password"
required/><br></br>
  <input type="number" id="phone" name="phone" placeholder=" Phone" required/>
  <input type="email" id="email" name="email" placeholder=" Email" required/><br></br>
  <input type="submit" class="uni-btn" value="Register">
  </form>
</div>
<div class="login_info">
<form action="UserLogPage.php" class="inputform">
<p style="margin-bottom:20px">Already an user?</p>
```

```
<input type="submit" class="uni-btn" value="Login">
</form>
</div>
</body>
</html>
```

B. Select Movie

```
<html>
<?php include('navbar1.html'); ?>
<link rel="stylesheet" href="css/styles.css">
<body>
<div class="form_display" id="grad">
<form action="" method="POST" id="movie_form">
<input type="date" name="date_select" class="desc_submit">
<br>

<select name="time_select" required>
<option value="choose time">Choose Time</option>
<option value="10:00:00"><time>10:00AM [10:00]</time></option>
<option value="13:00:00"><time>01:00PM [13:00]</time></option>
<option value="16:00:00"><time>04:00PM [16:00]</time></option>
<option value="19:00:00"><time>07:00PM [19:00]</time></option>
<option value="22:00:00"><time>10:00PM [22:00]</time></option>
</select>
<br>

<select name="movie_select">
<option value="No movie is selected">Choose Movie below</option>
 <?php
 foreach($names as $name){
   echo '<option value="'.$name.'">'.$name.'</option>';
 }
 ?>
</select>
<br>
<select name="seat_select" required>
```

```
<option value="No. of seats not selected">Choose no. of seats</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
</select>
<br>
<input type="submit" value="select" class="submit uni-btn">
</form>
</div>
<br>

<?php
$date_select = filter_input(INPUT_POST, 'date_select');
$time_select = filter_input(INPUT_POST, 'time_select');
$movie_select = filter_input(INPUT_POST, 'movie_select');
$seat_select = filter_input(INPUT_POST, 'seat_select');
?>

<div class="show_display" style="float:right">
<p>Selection will be displayed here</p>
<p>Movie: <?php echo $movie_select; ?> </p>
<p>Seats: <?php echo $seat_select; ?> </p>
<p>Date: <?php echo $date_select; ?> </p>
<p>Time: <?php echo $time_select; ?> </p>
<br>
<input type="button" class="uni-btn" value="Book" onclick="book_ticket()"
href="movie_selectQ.php" style="margin-left:60px">
</div>
```

```
<?php if(!empty($seat_select) && !empty($movie_select) && !empty($date_select) && !
empty($time_select))
{ ?>
 <script>
 book_ticket = function(){
 window.location="movie_selectQ.php"
 }</script>
<?php } ?>
</body>
</html>
```

C. Show Ticket

```
<html>
<link rel="stylesheet" href="css/styles.css">
 <body>
 <div class="ticket_display">
 <h1>Summary</h1>
 <table class="table_display">
 <tr class="table_items">
  <th>Name: </th>
  <td><?php echo $fullname; ?></td>
  <th>Seats: </th>
  <td><?php echo $no_seat; ?></td>
 </tr>
 <tr class="table_items">
  <th>Movie Name: </th>
  <td><?php echo $movie_name; ?></td>
  <th>Price: </th>
  <td><?php echo $net_amt; ?>/-</td>
 </tr>
 <tr class="table_items">
  <th>Date: </th>
  <td><?php echo $date; ?></td>
  <th>Price(<?php echo $no_seat; ?>): </th>
  <td><?php echo $amt_for_seats; ?>/-</td>
 </tr>
```

```
 <tr class="table_items">
   <th>Time: </th>
   <td><?php echo $time; ?></td>
   <th>Snacks Price: </th>
   <td><?php echo $snack_netamt; ?>/-</td>
 </tr>
 <tr class="table_items">
   <th>Snacks: </th>
   <td><?php echo $item. ' item(s)'; ?></td>
   <th>Net Amount: </th>
   <td><?php echo $total_amt; ?>/-</td>
 </tr>
</table>
<form action="ticket_showQ.php" method="POST">
<input type="submit" value="Proceed" class="uni-btn" style="margin:15px 0px 0px 0px;">
</form>
</body>
<html>
```

D. Add Movie

```
<!DOCTYPE html>
<?php include('Admin_navbar1.html'); ?>
<html lang="en" dir="ltr">
<link rel="stylesheet" href="css/styles.css">
 <head>
   <meta charset="utf-8">
   <title>Adding Movie</title>
 </head>
 <body>
<div class="add_form">
<h1>ENTER MOVIE DETAILS</h1>
 <form action="AddMovieP.php" method="POST" class="inputform"
enctype="multipart/form-data">
   <input type="text" id="Name" name="Name" placeholder=" Name" required><br></br>
   <input type="text" id="Genre" name="Genre" placeholder=" Genre" required><br></br>
   <input id="Language" name="Language" placeholder=" Language" required><br></br>
```

```
   <input id="ratings" name="Ratings" placeholder=" Ratings" required><br></br>
   <input id="Description" name="Description" placeholder=" Description" required><br></
br>
   <input id="Price" name="Price" placeholder=" Price" required><br></br>
   <input type="file" name="poster_img" id="filetype" style="font-size:14px"
required><br></br>
  <input type="submit" class="uni-btn" value=" Create">
</form>
</div>
</body>
</html>
```

E. Remove Movie

```
<html>
<?php include('Admin_navbar1.html'); ?>
<link rel="stylesheet" href="css/styles.css">
<body>
 <div class="delete_select">
          <form action="" method="POST"><br>
    <h1>SELECT THE MOVIE TO BE DELETED</h1>
    <select name="movie_del">
      <option value="No movie is selected">Choose Movie below</option>
      <?php
      foreach ($names as $name) {
        echo '<option value="' . $name . '">' . $name . '</option>';
      }
      ?>
    </select>
    <br></br>
    <input type="submit" value="Remove" class="uni-btn"><br>
  </form>
 </div>
```

F. Admin Page

```
<!DOCTYPE html>
<link rel="stylesheet" href="css/styles.css">
   <title>Admin Page</title>
  <body>
  <div id="Admin">
   <h1 style="color:white">Welcome, Admin.</h1>
    <div class="container-admin">
      <div class="left">
        <button type="button" name="button" class="button button1"><a
href="AddMovie.php" style="text-decoration:none;">Add new Movie</a></button>
      </div>
      <div class="right">
        <button type="button" name="button" class="button button2"><a
href="RemoveMovie.php" style="text-decoration:none;">Remove Movie</a></button>
        </div>
        <div class="mid">
        <button type="button" name="button" class="button button3"><a
href="booked_history.php" style="text-decoration:none;">Booked History</a></button>
        </div>
        <div class="mid-right">
        <button type="button" name="button" class="button button4"><a
href="change_default.php" style="text-decoration:none;">Change Defaults</a></button>
        </div>
     </div>
    </div>
  </body>
</html>
```

4.2 BACKEND CODE

The back-end, or the "server-side", is basically how the site works, updates and changes. This refers to everything the user can't see in the browser, like databases and servers. MySQL is an open-source 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 central component of the LAMP open-source web application software stack. LAMP is an acronym for "Linux ,Apache , MySQL, Perl/PHP/Python".

A. User Login

```
<?php
session_start();
require "db/dbcon.php";
$username = filter_input(INPUT_POST, 'username');
$password = filter_input(INPUT_POST, 'password');

$query = mysqli_query($conn, "SELECT * FROM user WHERE username = '$username'");
if(mysqli_num_rows($query) > 0){
  $row = mysqli_fetch_assoc($query);
  $user_db_pass = $row['pword'];

//decrypting the pass using global variables
global $pdo;
$pdo = new PDO('mysql:host=localhost;dbname=movie_db','root', '');
$key='kar';
$tmppass = $user_db_pass;

function mysql_aes_decrypt($tmppass, $key) {
  global $pdo;
  $stmt = $pdo->prepare("SELECT AES_DECRYPT(?, ?)");
  $stmt->execute(array($tmppass, $key));
  return $stmt->fetchColumn(0);
}
$decrypt_pass = mysql_aes_decrypt($tmppass, $key);
//end of pass decrypt
  if($password === $decrypt_pass){
    session_regenerate_id();

    $sql1 = mysqli_query($conn, "SELECT user_id FROM user WHERE
username='$username'");
    $row1 = mysqli_fetch_array($sql1);
    $_SESSION["session_user"]= $row1["user_id"];
```

```
    header('Location: movie_desc.php');
    exit;
  }
  else{
    // INCORRECT PASSWORD
    $error_message = "Incorrect Username or Password.";
  }
}
else{
  // Username NOT REGISTERED
  $error_message = "Incorrect Username or Password.";
}
?>
```

B. Select Movie

```
<?php
session_start();
include "db/dbcon.php";
$session_mov_id = $_SESSION["session_movie"];
$sql = mysqli_query($conn, "SELECT movie_id FROM movie WHERE
movie_name='$session_mov_id'");
$row = mysqli_fetch_array($sql);
$temp_mov_id= $row["movie_id"];

$_SESSION["session_movie_id"] = $temp_mov_id;

$temp_usr_id = $_SESSION["session_user"];
$seat_select = $_SESSION["session_seat"];
$date_select = $_SESSION["session_date"];
$time_select = $_SESSION["session_time"];
$combinedDT = date('Y-m-d H:i:s', strtotime("$date_select $time_select"));
$_SESSION["session_date_time"] = $combinedDT;

$query = mysqli_query($conn,"SELECT * FROM seats WHERE user_id = '$temp_usr_id'
AND movie_id = '$temp_mov_id' AND date_time = '$combinedDT'");
```

```
$row = mysqli_fetch_assoc($query);
if ($row>0) {

  echo "<script>
  alert('You already have this movie booked at the selected date and time');
  window.location.href='movie_select.php';
  </script>";
}
else {

$sql = "INSERT INTO seats(movie_id, user_id, date_time, no_seat)
VALUES('$temp_mov_id', '$temp_usr_id', '$combinedDT', '$seat_select')";

if (mysqli_query($conn, $sql)) {
  header("Location:snacks_select.php");
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);
}
?>
```

C. Show Ticket

```
<?php
session_start();
include "db/dbcon.php";
$temp_seat_id = $_SESSION["session_seat_id"];

$sql = "SELECT movie.movie_name, movie.net_amt, user.fullname, seats.date_time,
seats.no_seat, movie.net_amt , movie.net_amt*seats.no_seat AS amt_for_seats
FROM (((seats
INNER JOIN movie ON seats.movie_id = movie.movie_id)
INNER JOIN user ON seats.user_id = user.user_id)
INNER JOIN snacks on seats.seat_id = snacks.seat_id)
WHERE seats.seat_id = '$temp_seat_id'";
$result = mysqli_query($conn,$sql);
```

```
if (mysqli_num_rows($result) > 0) {
  // output data of each row
  while($row = mysqli_fetch_assoc($result)) {
    $movie_name = $row["movie_name"];
    $fullname = $row["fullname"];
    $combinedDT = $row["date_time"];
    $no_seat = $row["no_seat"];
    $net_amt = $row["net_amt"];
    $amt_for_seats = $row["amt_for_seats"];
    //spliting date and time
    $date = date('Y-m-d',strtotime($combinedDT));
    $time = date('H:i',strtotime($combinedDT));
    $day = date('d',strtotime($combinedDT));
  }
}
else {
  echo "0 results";
}
$sql1 = "SELECT SUM(snack_no) AS items FROM snacks WHERE seat_id =
'$temp_seat_id'";
$result1 = mysqli_query($conn,$sql1);
if (mysqli_num_rows($result1) > 0) {
  while($row1 = mysqli_fetch_assoc($result1)) {

    $item = $row1['items'];
  }
}

$sql2 = "SELECT snack_amt AS snacksamt FROM snacks WHERE seat_id =
'$temp_seat_id'";
$result2 = mysqli_query($conn,$sql2);
if (mysqli_num_rows($result2) > 0) {
  while($row2 = mysqli_fetch_assoc($result2)) {

    $snack_amt = $row2['snacksamt'];
```

```
  }
}
$snack_netamt = $snack_amt*$item;
$total_amt = $snack_netamt + $amt_for_seats;

$sql3 = "INSERT INTO ticket(seat_id, total_net_amt) VALUES('$temp_seat_id',
'$total_amt')";
 $result3=mysqli_query($conn, $sql3);
 if($result3){
  echo "<script>
  alert('Ticket booked successfully');
  window.location.href='movie_desc.php';
  </script>";
 }
else{
  echo "Error: ". $sql3 ."". $conn->error;
}
$conn->close();
?>
```

D. Add Movie

```
<?php
session_start();
require "db/dbcon.php";
$Name = filter_input(INPUT_POST, 'Name');
$Language = filter_input(INPUT_POST, 'Language');
$Genre = filter_input(INPUT_POST, 'Genre');
$Ratings = filter_input(INPUT_POST, 'Ratings');
$Description = filter_input(INPUT_POST, 'Description');
$Price = filter_input(INPUT_POST, 'Price');

$sql = "INSERT INTO movie(movie_name, movie_lang, movie_genre, ratings, movie_des,
price)
 VALUES('$Name', '$Language', '$Genre','$Ratings','$Description','$Price')";
$result = mysqli_query($conn, $sql);
```

```
$filename = $_FILES["poster_img"]["name"];
$tempname = $_FILES["poster_img"]["tmp_name"];
$folder = "posters/" . $filename;

// Get all the submitted data from the form
$sql2 = "INSERT INTO poster(movie_name, poster_img)
    VALUES ('$Name','$filename')";

if (mysqli_query($conn, $sql2)) {

// Now let's move the uploaded image into the folder: image
   if (move_uploaded_file($tempname, $folder)) {
      $msg = "Image uploaded successfully";
   } else {
      $msg = "Failed to upload image";
   }
}
$result2 = mysqli_query($conn, "CALL TAX_CALC()");
if ($result2) {

   echo ("<script LANGUAGE='JavaScript'>
   window.alert('Succesfully Updated');
   window.location.href='AdminPage.php';
   </script>");
}
mysqli_close($conn);
```

E. Remove Movie

```
<?php
$movie_del = filter_input(INPUT_POST, 'movie_del');
  if (isset($movie_del)) {

$sql1 = "SET FOREIGN_KEY_CHECKS=0;";
// disable foregin key check
mysqli_query($conn, $sql1);
$sql2 = "DELETE FROM movie WHERE movie_name='$movie_del'";
```

```
if (mysqli_query($conn, $sql2)) {
  $sql3 = "DELETE FROM poster WHERE movie_name='$movie_del'";
  mysqli_query($conn, $sql3);

   echo ("<script LANGUAGE='JavaScript'>
   window.alert('Movie Removed Successful');
   window.location.href='AdminPage.php';
   </script>");
   }
   else {
     echo "Error: " . $sql2 . "<br>" . mysqli_error($conn);
   }
   $sql4 = "SET FOREIGN_KEY_CHECKS=1;";
   //Re-enable foregin key check
   mysqli_query($conn, $sql4);
 }
```

F. Booked History

```
<?php
session_start();
include "db/dbcon.php";
include('Admin_navbar1.html');

$sql = "SELECT movie.movie_name, user.username, seats.date_time, seats.no_seat,
seats.seat_id
FROM ((seats
INNER JOIN movie ON seats.movie_id = movie.movie_id)
INNER JOIN user ON seats.user_id = user.user_id)";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
       // output data of each row
  while ($row = mysqli_fetch_assoc($result)) {

        $combinedDT = $row["date_time"];
```

```
        $movie_name = $row["movie_name"];
        $seat_id = $row["seat_id"];
        $user_name = $row["username"];
        $no_seats = $row["no_seat"];
        $date = date('y-m-d', strtotime($combinedDT));
        $time = date('H:i', strtotime($combinedDT));
    }
}
else {
     echo "0 results";
     }
$conn->close();
```

4.3 DB ACCESS AND CONNECTIVITY

A connection needs to be established between the front end and back end, for retrieving the desired data from the database to the end user through the GUI. This connection is done using MySQLi api.

MySQLi

It is an API that uses a connector function to link the backend of the PHP app to the MySQL database. It provides a better set of functions and extensions. It works just like the previous version, but it is safer and faster.

4.4 CONNECTION CODE

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "movie_db";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname)
// Check connection
if ($conn->connect_error) {
  die("" . $conn->connect_error);
}
```

4.5 TRIGGERS

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. It is mostly used to maintain the integrity of the information on the database.

```
DELIMITER $$
CREATE TRIGGER `pass_crypt` BEFORE INSERT ON `user` FOR EACH ROW SET
new.pword := AES_ENCRYPT(new.pword,'kar')
$$
DELIMITER ;
```

```
DELIMITER $$
CREATE TRIGGER `reset_pass_crypt` BEFORE UPDATE ON `user` FOR EACH ROW
SET new.pword := AES_ENCRYPT(new.pword,'kar')
$$
DELIMITER ;
```

4.6 STORED PROCEDURES

A stored procedure is used to check whether the user is valid and returns their user ID when they log in. It is used in add, delete and search procedures. The procedure takes animal id as input parameter. It checks the presence of such user in the database. If present, it fetches the corresponding animal ID and returns it as an output parameter. This user ID is maintained during the session to keep track of the user activity.

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `TAX_CALC`()  BEGIN
DECLARE ch_done INT DEFAULT 0;
DECLARE C_A INT(10);
DECLARE C_B INT(10);
DECLARE C_C INT(10);
DECLARE C_net_amt DEC(10,2);
DECLARE C_tax DEC(10,2);

DECLARE C_Tprice CURSOR FOR
SELECT `price` AS A,`movie_id` AS B,`tax` AS C FROM `movie` WHERE `net_amt` IS
NULL
FOR UPDATE;
```

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET ch_done = 1;

OPEN C_Tprice;
FETCH C_Tprice INTO C_A,C_B,C_C;

SET C_tax = (C_C/100);
SET C_net_amt=C_A + (C_A*C_tax);
UPDATE `movie` SET `net_amt`=C_net_amt where `movie_id`=C_B;
CLOSE C_Tprice;
IF(ch_done = 1) THEN

        SELECT 'Oh no!'; -- handle the No data error!
   END IF;
END$$
DELIMITER ;
```

SNAPSHOTS

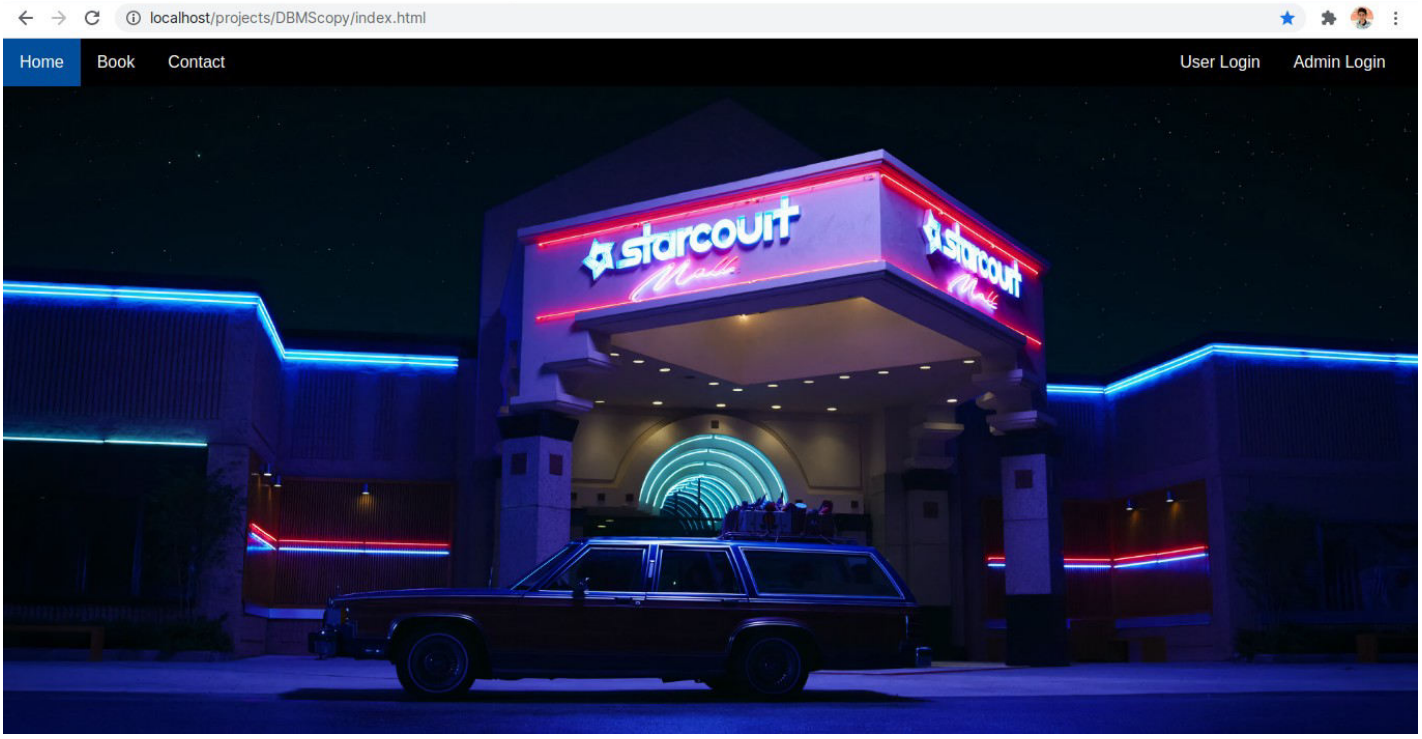

Fig 5.1: Index Page

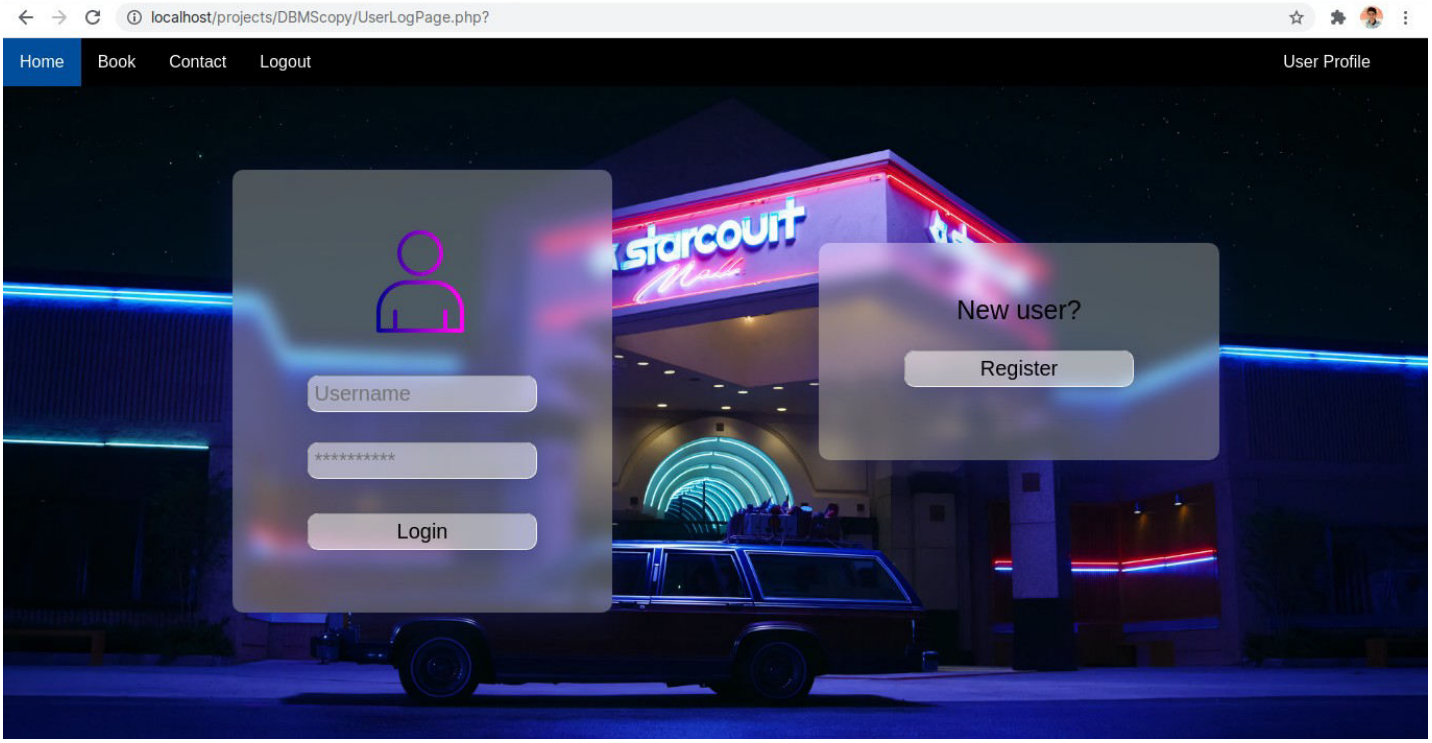

Fig 5.2: User Login Page

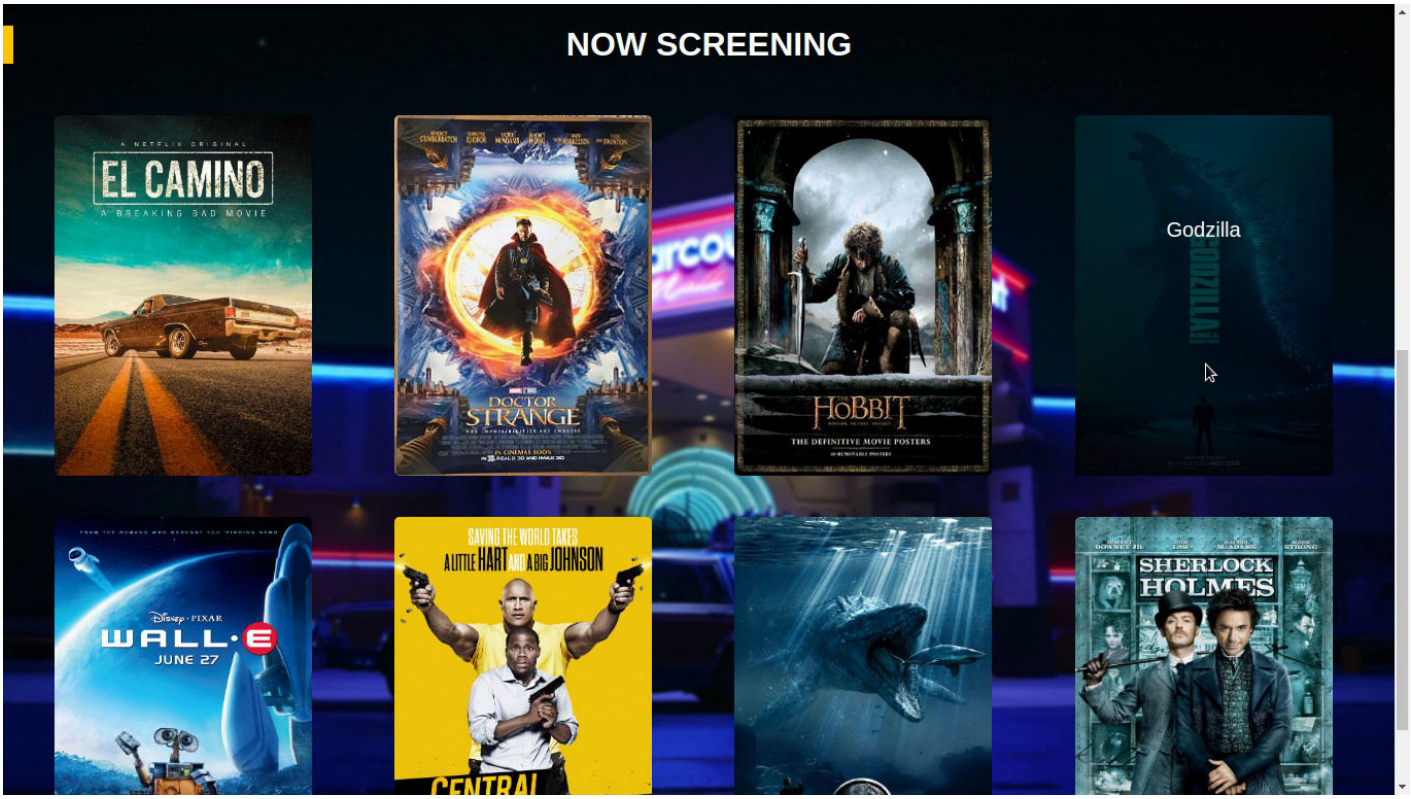

Fig 5.3: Movies Now Screening

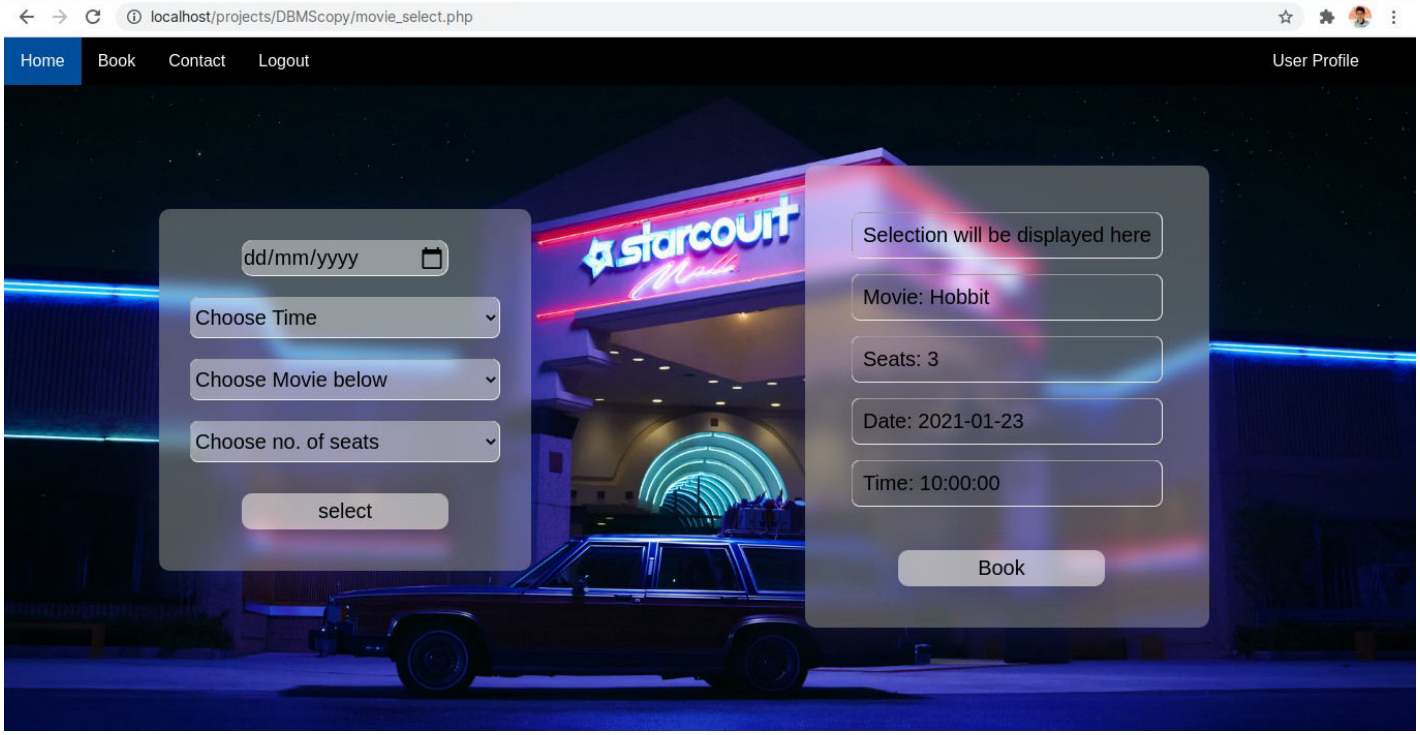

Fig 5.4: Select Movie

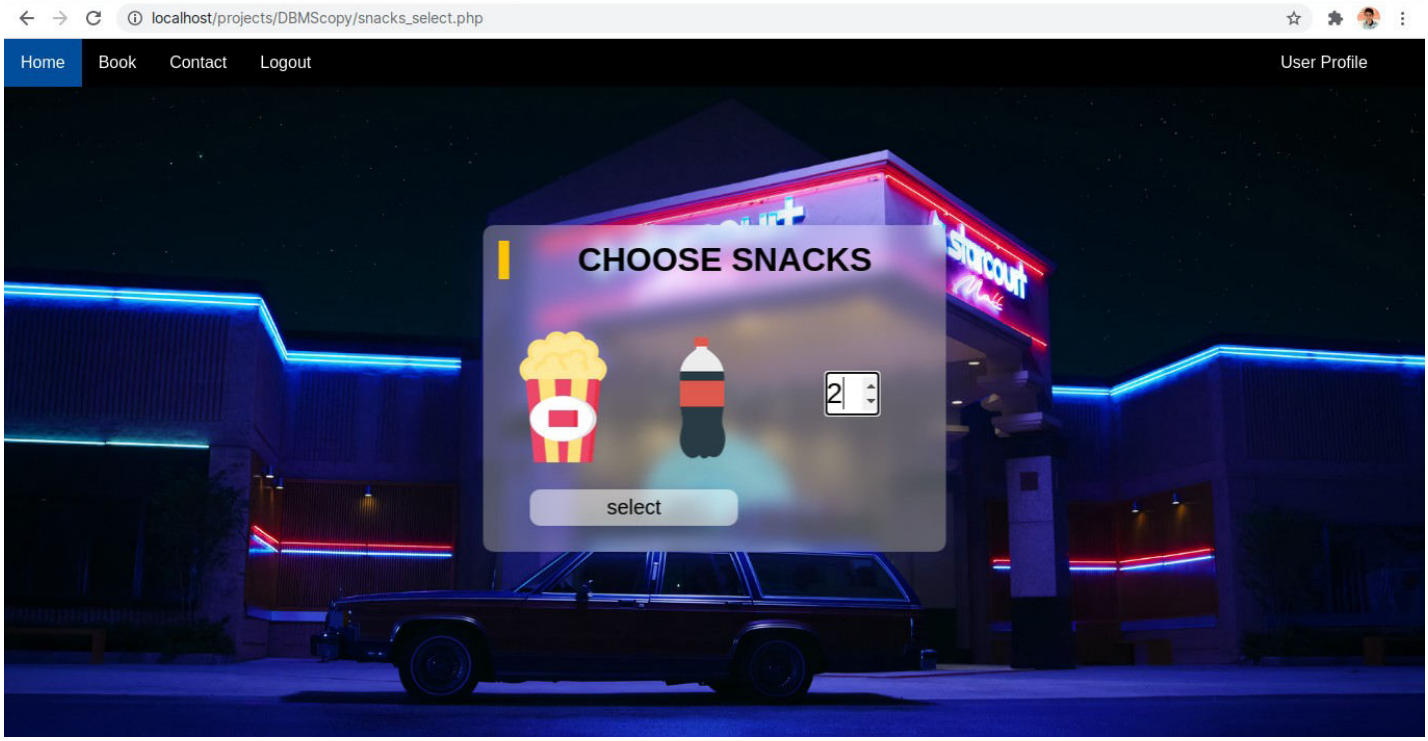

Fig 5.5: Select Snack

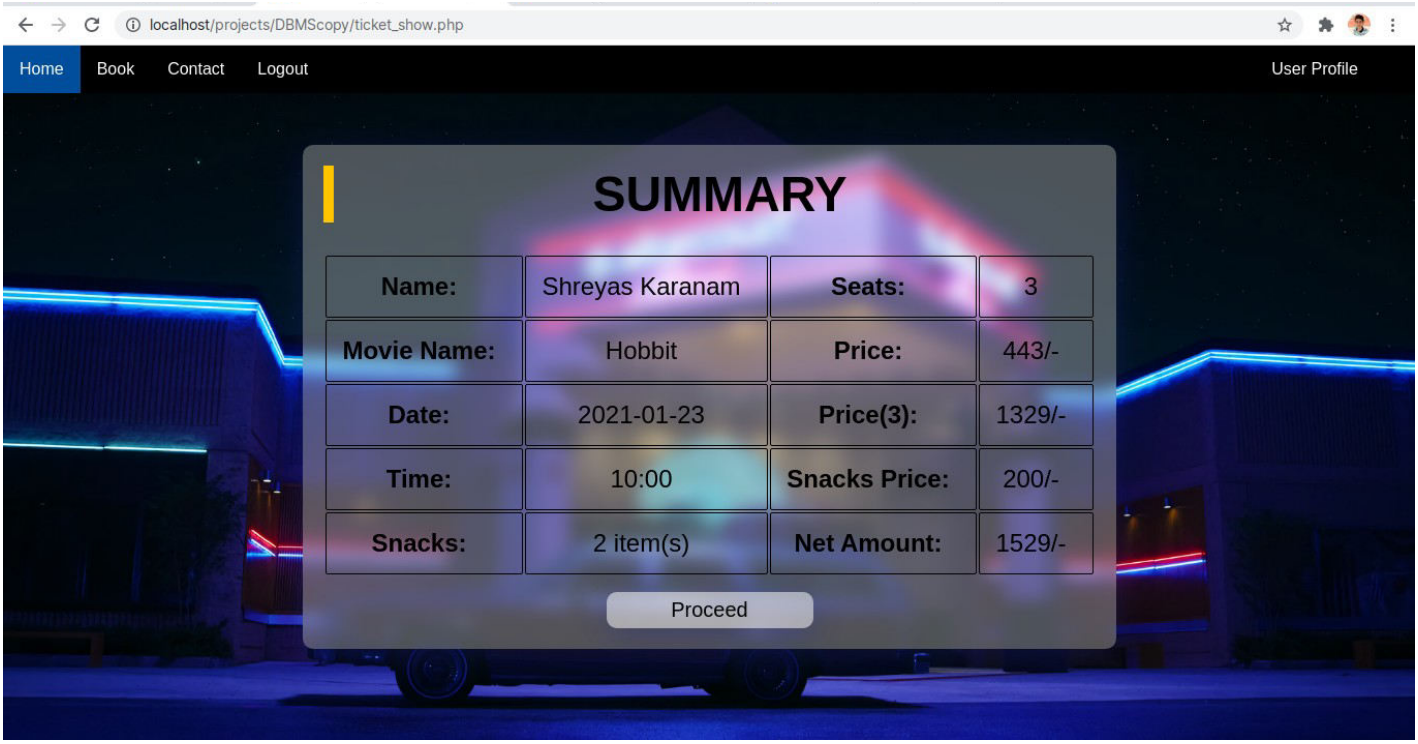

Fig 5.6: Ticket Summary

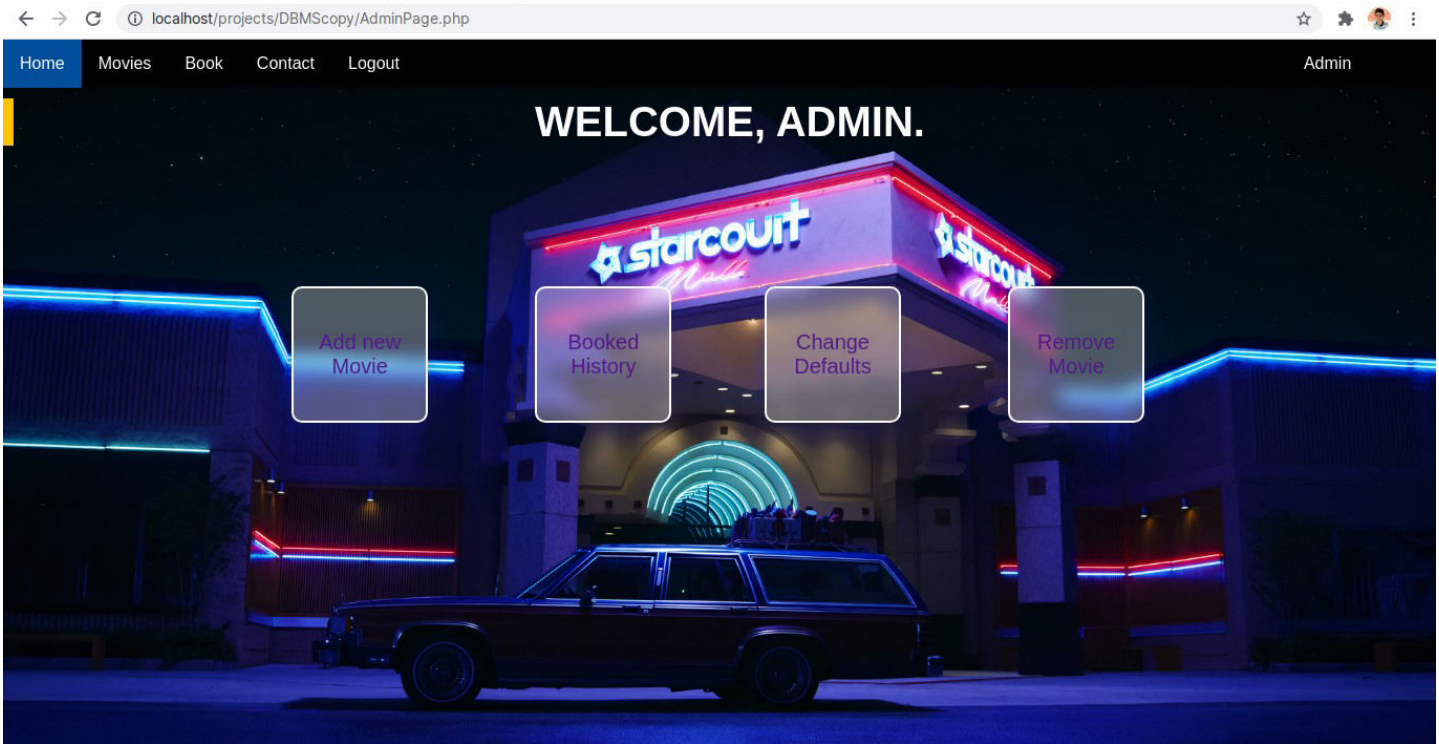

Fig 5.7: Admin Page

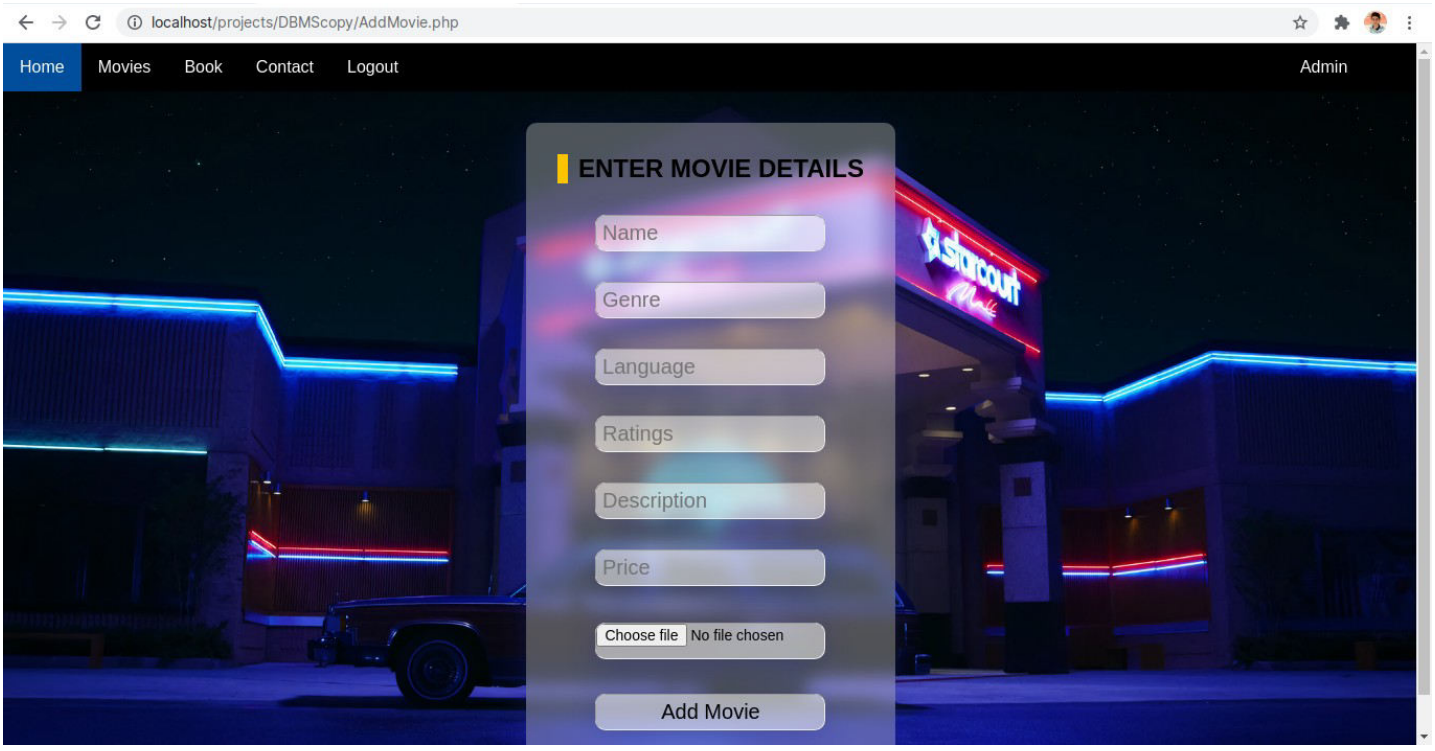

Fig 5.8: Add Movie

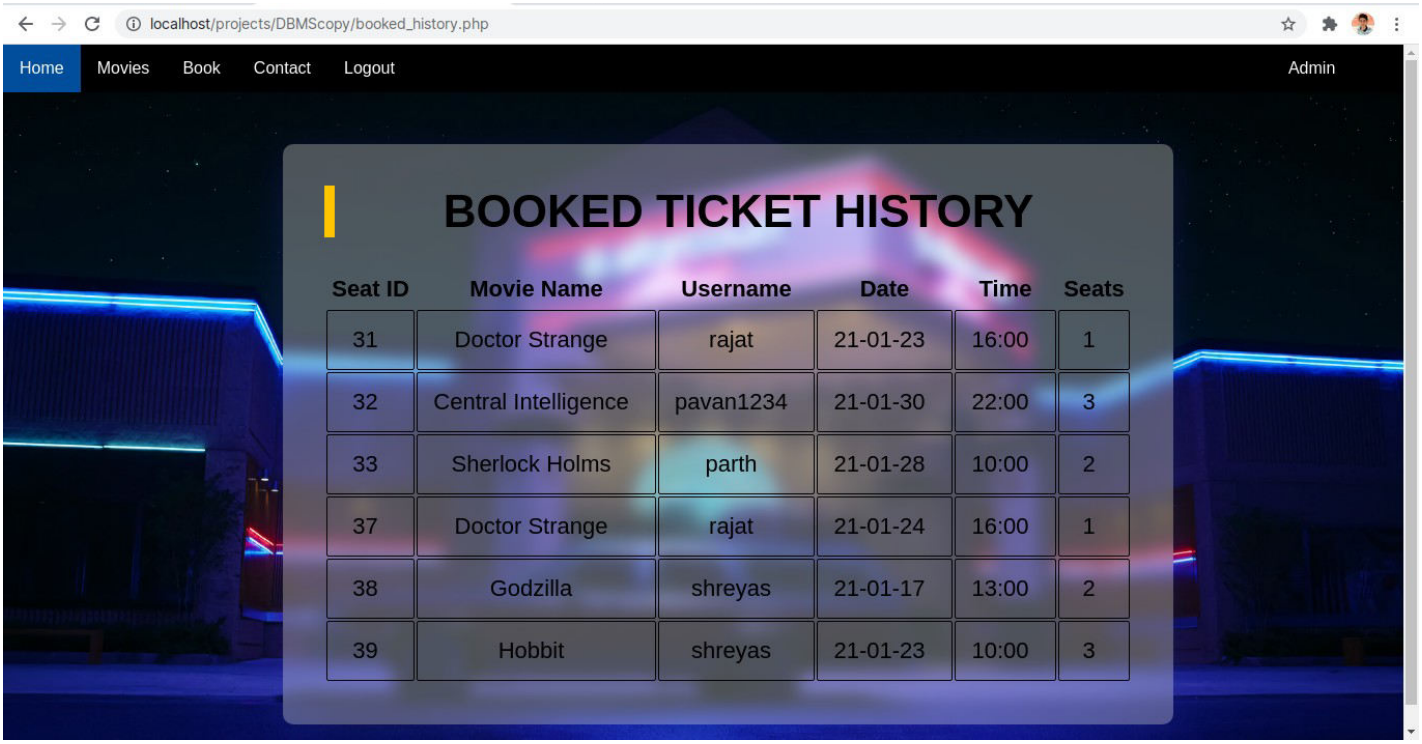

Fig 5.9: Booked history

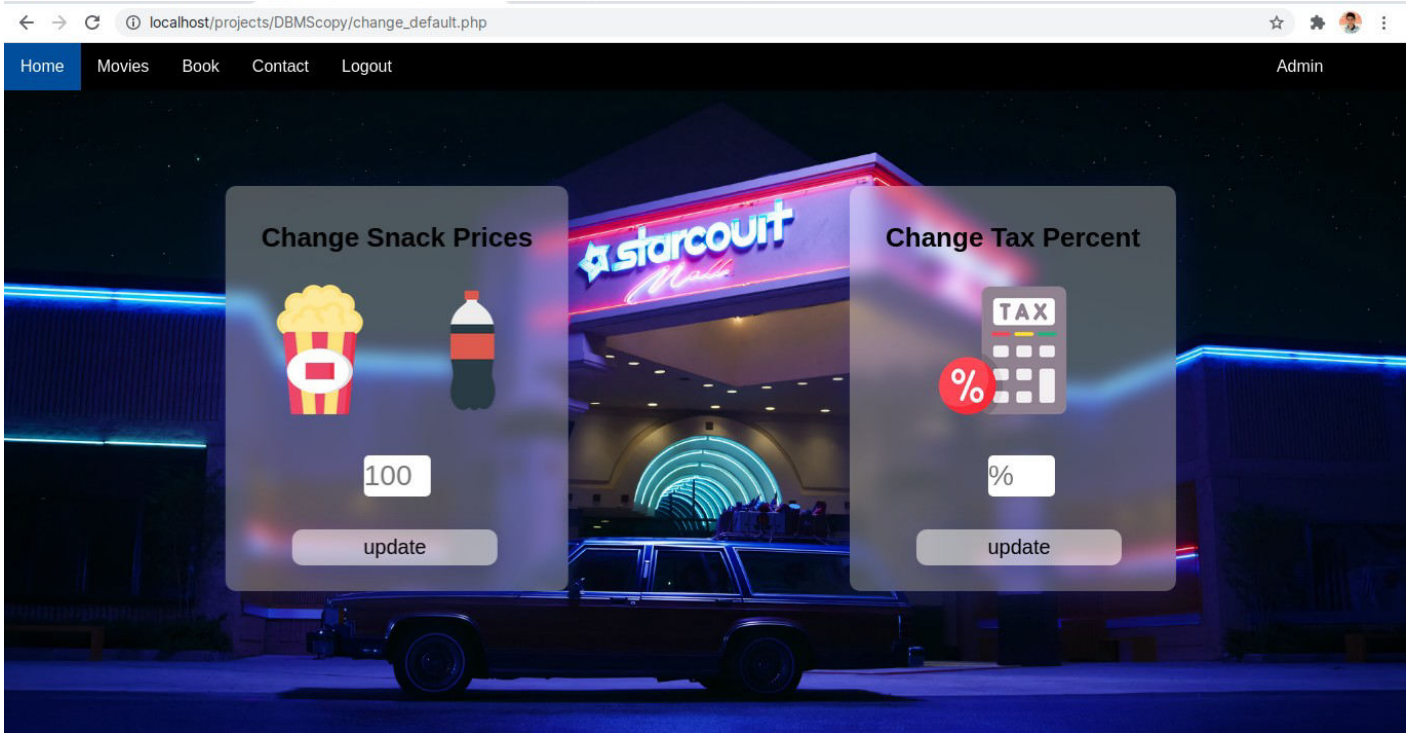

Fig 5.10: Update Defaults

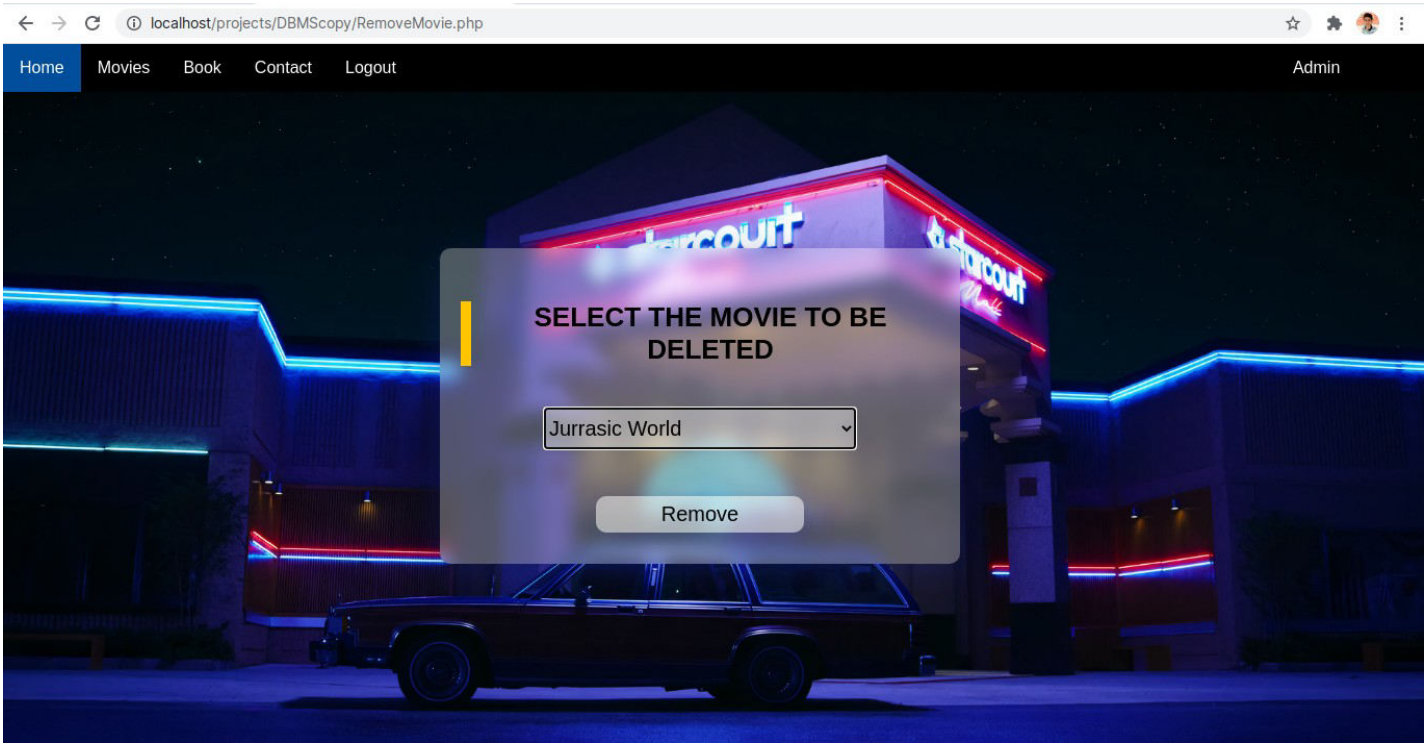

Fig 5.11: Delete Movie

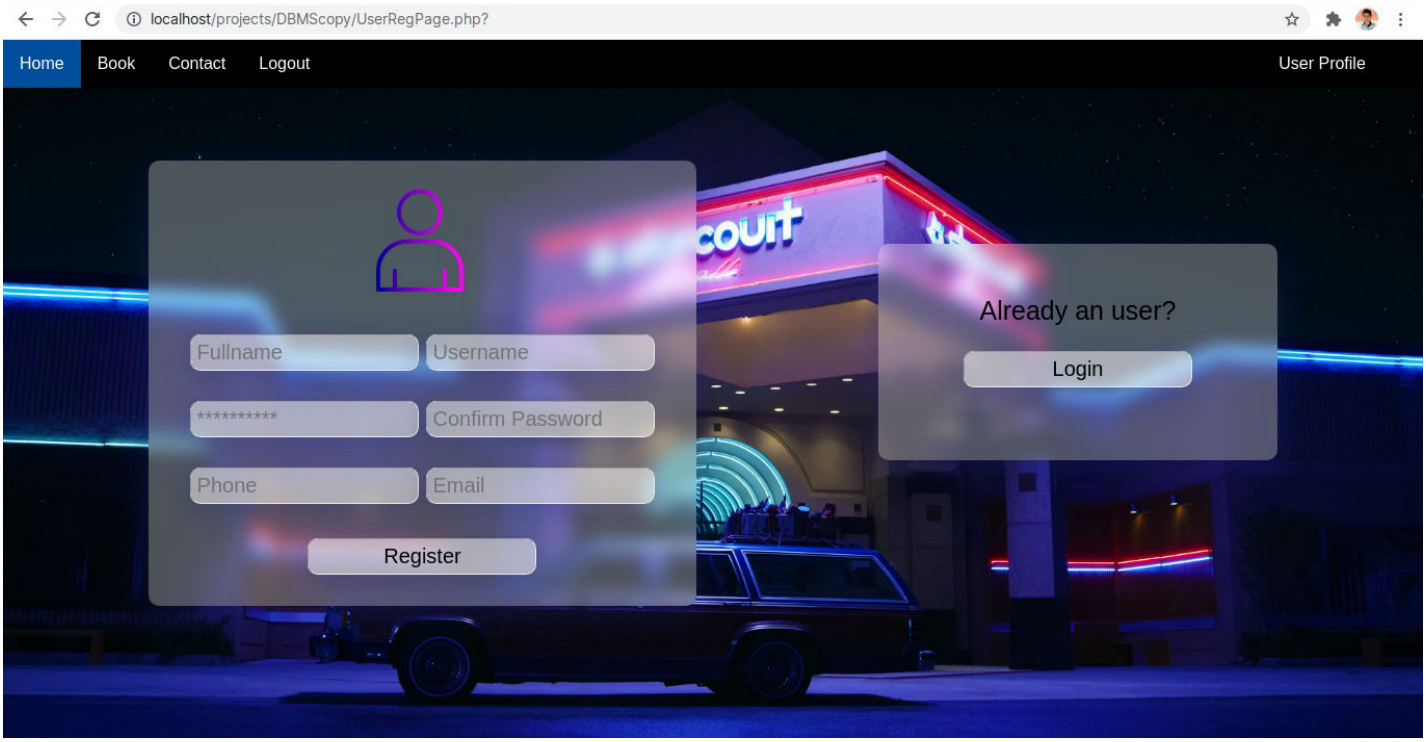

Fig 5.12: User Registration

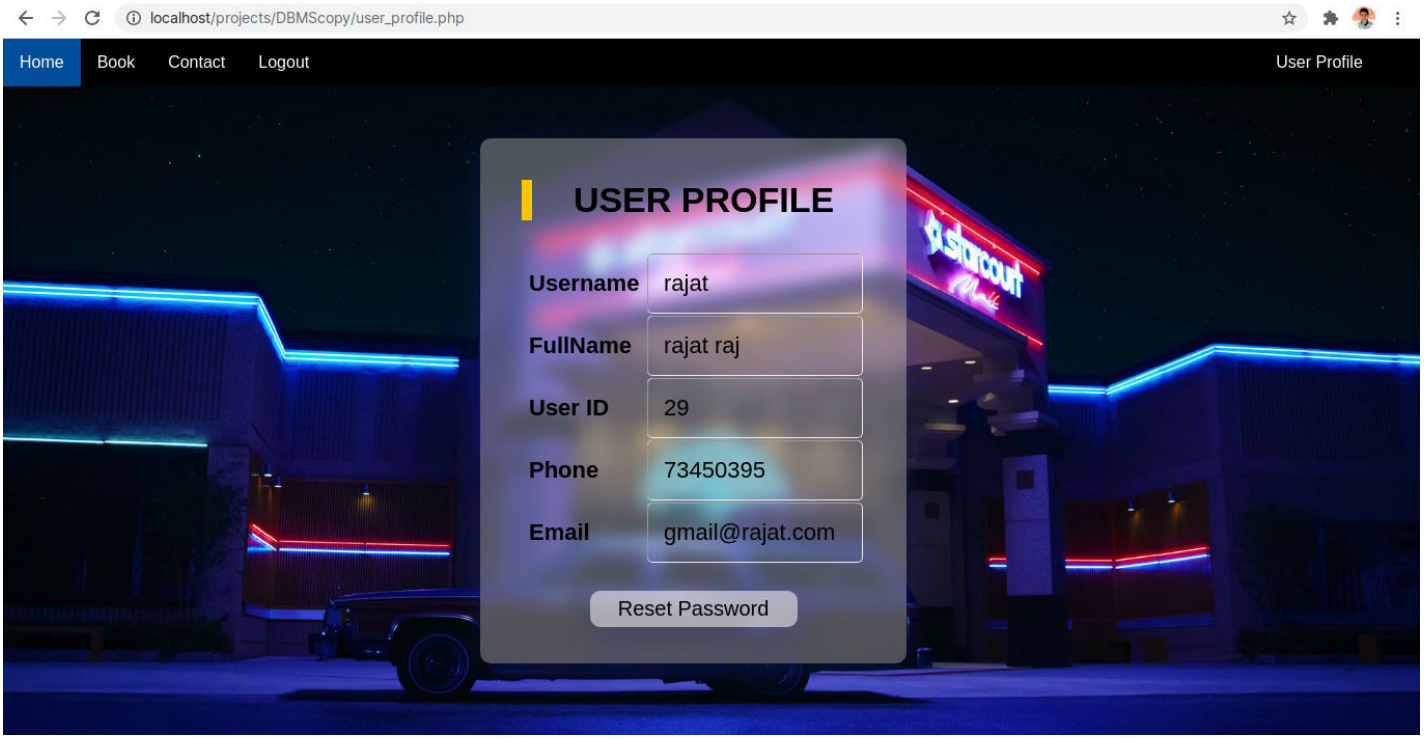

Fig 5.13: User Profile

CONCLUSION

This project gave us the idea about, how large data are stored inside a database and organized so that it can be retrieved easily and in a more efficient way. It also helped us to create application using HTML and CSS code and connecting the back end with front end using the PHP code, so that any actions that are performed in the front end are reflected in the backend and also any modifications made at the back end can also be seen in the front end. It also gave us complete idea about how the queries retrieve data from multiple tables and the working of structured procedure and the triggers. This project is used to maintain Movie Ticket Booking.

Future Enhancement

In future, this system can be extended to the desired level so that it can adapt to changing technology and enhance its performance. The latest data can also be included into the database, so that the future retrieval keeps all the necessary data that has been updated. The system can be provided with any sort of queries to perform the required actions with the help of the schema. It also performs some computations that can be altered or modified as per the latest requirements.

REFERENCES

[1] Stack overflow
http://stackoverflow.com

[2] W3Schools
https://www.w3schools.com

[3] Fundamentals of Database Systems, 7th Edition
Ramez Elmasri, University of Texas at Arlington
Shamkant B. Navathe

[4] Icons
https://iconscout.com

[5] Google / Google Images
www.google.com

[6] YouTube
www.youtube.com

[7] Booking sites
https://in.bookmyshow.com