Chapter 1: Introduction

This specific work concerns an information system for a cinema page where a visitor to the page can view the available movies. They can also request the creation of an account, which needs to be approved by an administrator. The user of the page can also view the movies and book tickets for them by selecting the number of tickets and a screening. The system administrator can add users, approve, edit, and delete users, as well as perform the same actions for movies, screenings, and reservations. This system is implemented using PHP, JavaScript, CSS, and SQL.

Chapter 2: Description of the Implemented System

For this particular work, a network-centric information system was developed that satisfies the requested requirements. Specifically, it is a cinema system where the following can be done:

1) User registration: A user can sign up, and after an admin reviews their request, they can log in to the page. Before that, they can only browse movies and view information about them.

2) System user management: Once there are users and requests from users, the admin can access admin.php and manage these users. More specifically, they can a) approve requests for account creation, b) update user information, and c) delete a user.

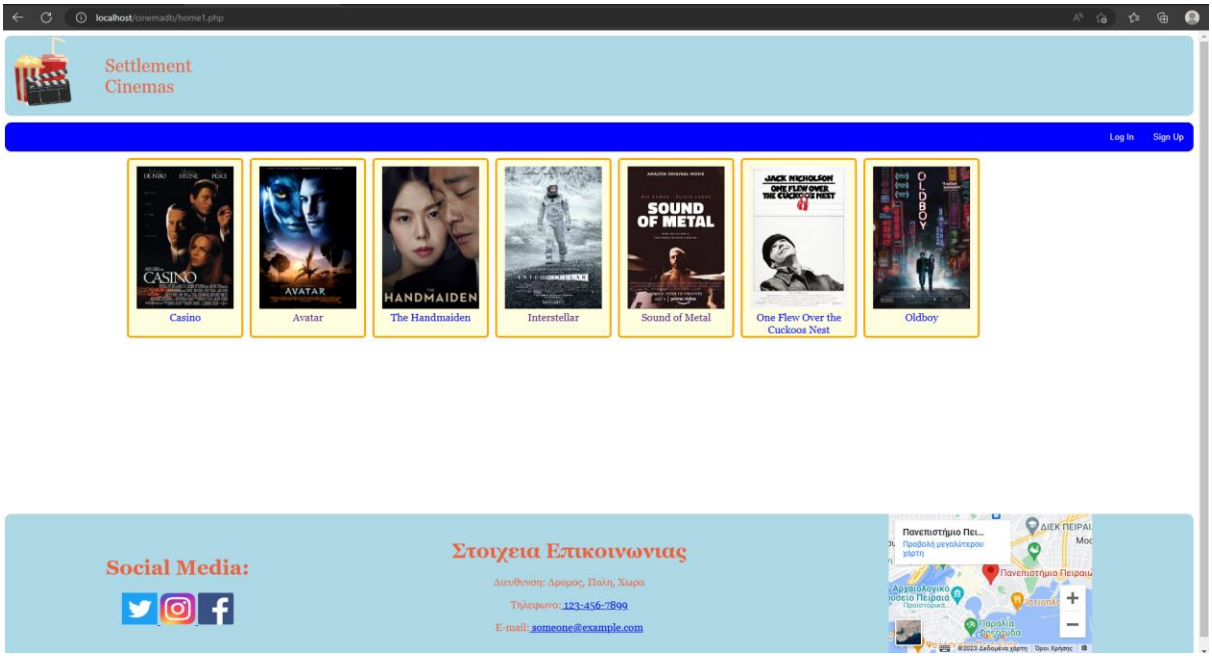
3) Management of structural elements of the system: The admin has access to all the elements offered to users and created by them (reservations). Specifically, they can create and delete movies, create and delete screenings, and edit all of these.

4) Browsing products: A regular user can view the movies available at the cinema without being logged in and click on them for more information about the movie.

5) Product reservation: A logged-in user, i.e., someone approved by the admin, can book tickets for a movie, creating a reservation for that movie. The admin can also view all reservations.

Chapter 3: System Implementation

Initially, a user is on home1.php. There they can see the movies currently available at the cinema.

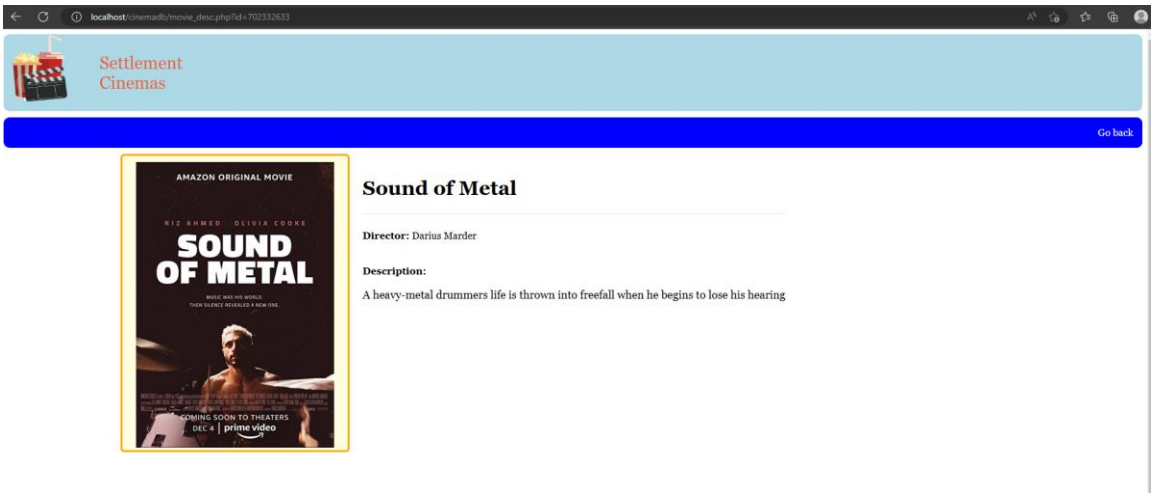


The movies are displayed as follows: an SQL statement is executed and passed to $result, and through a while loop using fetch_array, each row of $result is passed to $row, representing a row of the specific table from the database. Then, the movie images are displayed in a specific way based on the movie_name (for this reason, the image names in the images folder should be the same as in the database). Additionally, through href=movie_desc?id=$row['movie_id'], we are redirected to the specific PHP file with the movie's code passed in the URL.

```php
<?php
$sql='select * from movie';
$result=$conn->query($sql);
if (!$result) die($conn->error);

// Εμφανιση ταινιων και redirect στο movie_desc.php αν πατηθει μια ταινια
while ($row=$result->fetch_array(MYSQLI_ASSOC)){
    echo "<div class='movies' style='border: 3px solid orange; width: 180px; height: auto; text-align: center; background-color: lightyellow; border-radius: 6px; ma
    <div class='item'><a href='movie_desc.php?id=".$row['movie_id']."' style='text-decoration: none'><img src='images/$row[movie_name].png' width=150 height=220 sty
}
?>
```

Movie Monitoring

After being redirected to movie_desc.php, further information about the movie is available.




```
<form class='movies' method='post' action='tickets.php?id=$movie_id'>
<div class='container'>
<h1>".$movie_name."</h1>
<hr>



<label for='user_name'><b>Director: </b>".$movie_director."</label>

<br>
<br>
<br>

<label for='user_name'><b>Description: </b><p style='font-size: 18px'>".$movie_desc."</p></label>
```

The way the movie ID is obtained for the movie we have clicked on is through the GET operation. Thus, an SQL statement is executed to retrieve the information for the specific movie we clicked on.

```php
<?php require_once('common.php');
session_start();

////////// MOVIE
    // pairnei apto url to id kai me vasi auto ftiaxnei kati variables pou ta xrhsimopoioume kato gia na emfanisoume alles plirofories
    if (isset($_GET['id'])) {
        $movie_id = $_GET['id'];
        $sql = "SELECT * FROM movie WHERE movie_id = $movie_id";
        $result2 = $conn->query($sql);

        if ($result2->num_rows > 0) {
            $row = $result2->fetch_assoc();
            $movie_name = $row["movie_name"];
            $movie_director = $row["movie_director"];
            $movie_desc = $row['movie_desc'];

        }
    }

?>
```

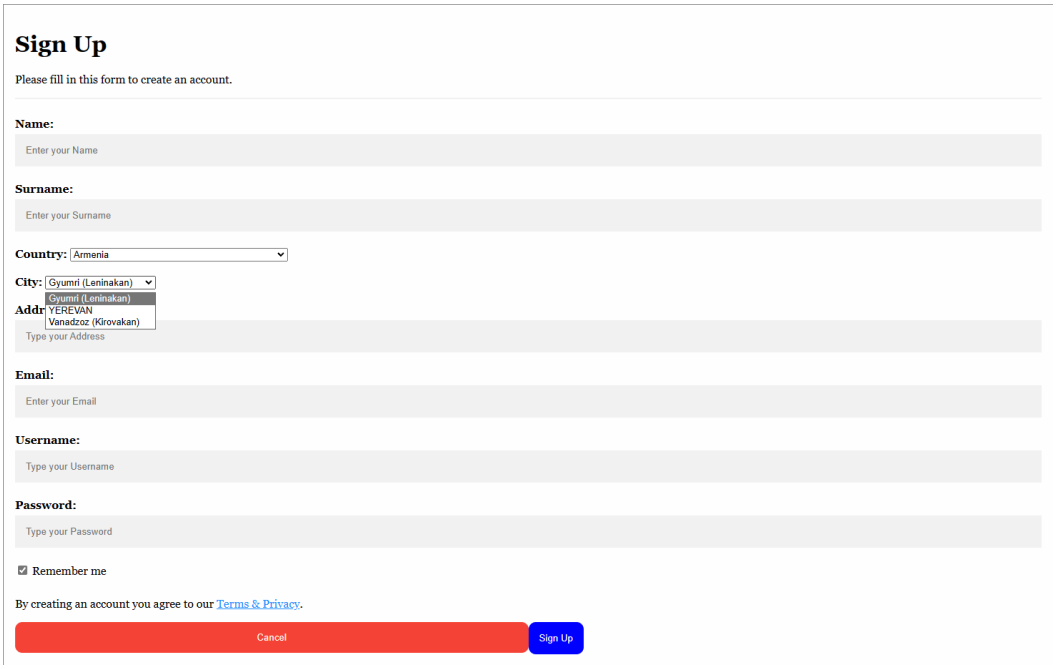
Additionally, there is a button labeled "Go back" that takes you back to home1.php. The same button exists in other PHP files to log out, but it redirects to log_out.php (which will be shown below).

```
<div id="signup">
    <li class="user_welcome">
    </li>
    <li >
    <!-- Log out button που σε πεταει στο home1.php -->
    <a href="home1.php">Go back</a>
</li>
</div>
```

Sign Up

In home1.php, a visitor has the option to sign up.

By clicking this button, a Modal will appear with a form where they can enter their details.



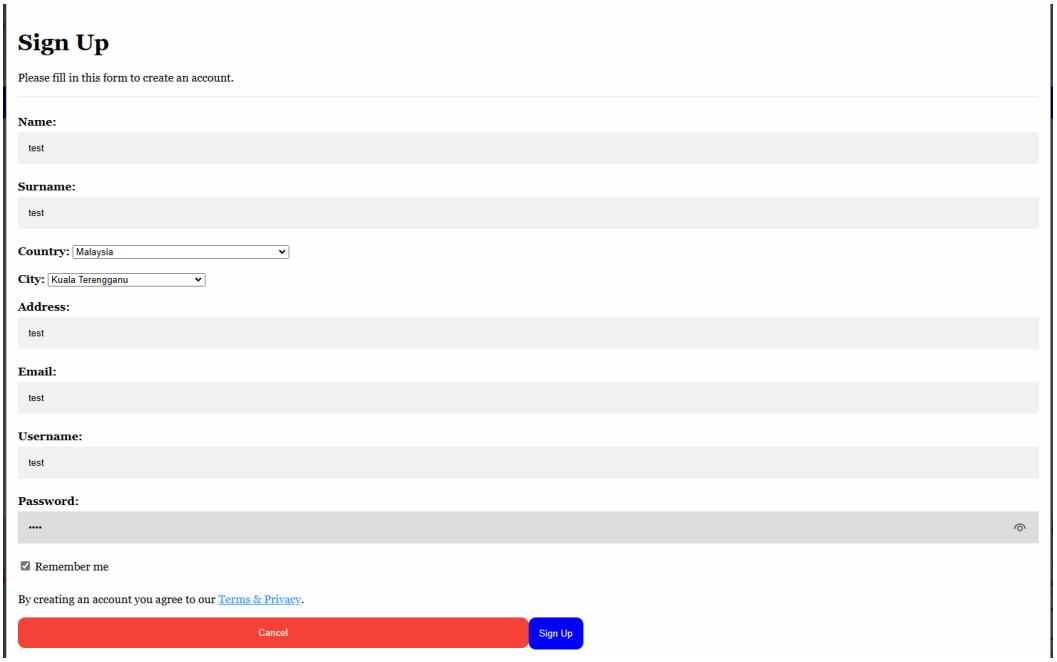
The Country and Cities fields have available values displayed in dropdown list boxes through an API. Depending on the selected country, the available cities for that country are provided by the API. (Note: The implementation of this specific API may have a different number of cities for each country, resulting in varying numbers of cities displayed for each country.) Specifically, for the country selection, a function in the script.js file runs, displaying the countries, and then populating the options in the cities dropdown box.

```html
<label for="countries"><b>Country:</b></label>
<select onmouseup="showCities();" class="text" name="user_country" id="countries" ></select>

<br>
<br>

<label for="cities"><b>City:</b></label>
<select class="text" name="user_city" id="cities"></select>
```

```javascript
//++++++++    Rest API για τις χωρες
const xhttp2 = new XMLHttpRequest();
const select2 = document.getElementById("countries");
const flag = document.getElementById("flag");

let countries;

xhttp2.onreadystatechange = function(){
    console.log('this.status', this.status);
    if (this.readyState == 4 && this.status ==200){
        countries = JSON.parse(xhttp2.responseText);
        assignValues2();
        //handleCountryChange();
    }
};
```

```javascript
//++++++++  Rest API για τις πολεις
function showCities(){

    const xhttp = new XMLHttpRequest();
    const select = document.getElementById("cities");
    const selectedCountry = document.getElementById("countries").value;

    let cities;

    xhttp.onreadystatechange = function(){
        console.log('this.status', this.status);
        if (this.readyState == 4 && this.status ==200){
            cities = JSON.parse(xhttp.responseText);
            assignValues();
            //handleCountryChange();
        }
    };
    xhttp.open("GET", "https://countriesnow.space/api/v0.1/countries/population/cities", true);
    xhttp.send();

    select.length=0;
    function assignValues(){
        cities["data"].forEach(data => {
            if(data.country==selectedCountry){
            const option = document.createElement("option");
            console.log('data', data)
            option.value = data.city;
            option.textContent = data.city;
            select.appendChild(option);
            }
        });
    }
```

Once all the fields in the sign-up form are filled out, an SQL statement is executed to insert the corresponding data into the user table.




```php
<?php require_once('common.php');

session_start();
$_SESSION;

function random_num(){
    $var = rand(1,9999999999);
    return $var;
}

    $user_role ="not_assigned";

        // Sign_up ενος user
        if (isset($_POST['user_name'])&&isset($_POST['user_surname'])&&isset($_POST['user_country'])&&isset($_POST['user_city'])&&isset($_POST['user_address'])&&isset($_POST['
            $user_id= random_num();
            $user_name=$_POST['user_name'];
            $user_surname=$_POST['user_surname'];
            $user_country=$_POST['user_country'];
            $user_city=$_POST['user_city'];
            $user_address=$_POST['user_address'];
            $user_email=$_POST['user_email'];
            $user_username=$_POST['user_username'];
            $user_password=$_POST['user_password'];


            $query="INSERT into user values"."('$user_id','$user_name', '$user_surname', '$user_country', '$user_address', '$user_email', '$user_username', '$user_password', '$
            echo ($query);
            $result = $conn->query($query);

            if(!$query) echo ("failed").$conn->error."<br><br>";
        }
```

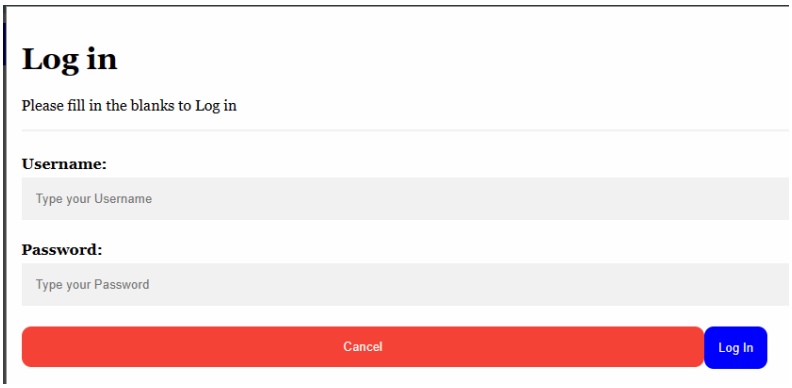
INSERT into user values('6991912249','test', 'test', 'Malaysia', 'test', 'test', 'test', 'test', 'not_assigned', 'Kuala Terengganu')

☐	Επεξεργασία	Αντιγραφή	Διαγραφή	6991912249	test	test	Malaysia	test	test	test	test	not_assigned	Kuala Terengganu

When a sign-up occurs, the default value for the user_role is "not_assigned." A user with this user_role cannot log in yet.

Log in

When a user's registration has been approved and the user_role has changed from "not_assigned" to "User," they can log in. By clicking the log-in button, a modal form appears.



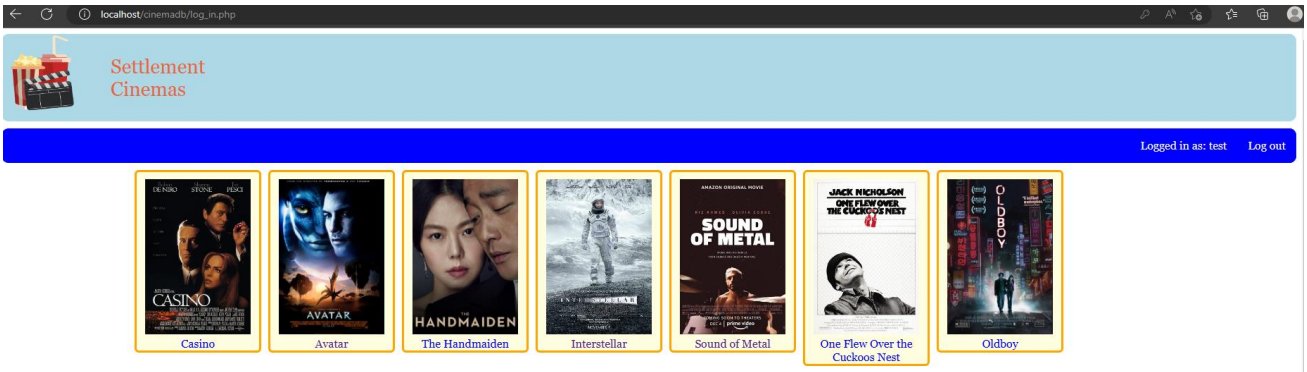

```
<!-- LOGIN -->
<div id="id02" class="modal">
    <span onclick="document.getElementById('id02').style.display='none'" class="close" title="Close Modal">X</span>
    <form class="modal-content" action="home1.php" method="post">
        <div class="container">
        <h1>Log in</h1>
        <p>Please fill in the blanks to Log in</p>
        <hr>


    <!-- log in Modal -->
        <label for="user_username"><b>Username:</b></label>
        <input type="text" placeholder="Type your Username" name="log_username" required>

        <label for="user_password"><b>Password:</b></label>
        <input type="password" placeholder="Type your Password" name="log_password" required>

        <div class="clearfix">
            <button type="button" onclick="document.getElementById('id02').style.display='none'" class="cancelbtn">Cancel</button>
            <button type="submit" class="signup">Log In</button>
        </div>
        </div>
    </form>
</div>
```

After filling in the blanks and clicking the log-in button, the entered data is stored in the "log_username" and "log_password" variables. Then, an SQL statement is executed to retrieve a user with the same user_username as the "log_username." The database is checked to see if such a record exists, and the result is stored in the "user_data" variable. Next, a check is performed to see if the "log_password" matches the user_password of that specific user in the database. If it does, a SESSION is created with the user_id. This creates a super global variable that helps determine if a user is logged in when performing functions that require being logged in, such as booking tickets.

```php
session_start();
$_SESSION;
```

```php
// Ελεγξε αν στο log-in modal εχει τιμες
if (isset($_POST['log_username'])&&isset($_POST['log_password'])){
    $log_username = $_POST['log_username'];
    $log_password = $_POST['log_password'];

    $query2 = "select * from user where user_username = '$log_username' limit 1";
    $result2 = mysqli_query($conn, $query2);

    if($result2)
    {
        // Ελεγχει αν υπαρχει η εγγραφη στο database
        if($result2 && mysqli_num_rows($result2) > 0)
        {
            // Περναει τα αποτελεσματα στην μεταβλητη $user_data
            $user_data  = mysqli_fetch_assoc($result2);

            // Ελεγχει αν το password του user ειναι αυτο που εχει το modal
            if($user_data['user_password'] === $log_password)
            {
                // Δημιουργει το Session με το id του user που εκανε log in
                $_SESSION['user_id'] = $user_data['user_id'];

                header("Location: log_in.php");
                die;
            }

        }
    }
    // Σε περιπτωση που μπουν λαθος στοιχεια
    echo "<script>alert('Wrong Username or Password!')</script>";


}
```

After a user successfully logs in, they are redirected to log_in.php.



Log out

When the log-out button is pressed, we go to the log_out.php file.

log_out.php

```php
<?php
    session_start();

    //Διακοπτει το SESSION και σε πεταει στο home1.php
    if(isset($_SESSION['user_id']))
        {
            unset($_SESSION['user_id']);
        }
    header("Location: home1.php");
    die;
?>
```

There, the SESSION is terminated, preventing access to other PHP files even if they are accessed via the URL.

Buy tickets

When the log-in occurs, the check_login() function is executed. It checks if the user who logged in has a pre-existing SESSION. If not, they are redirected back to home1.php. Additionally, if the user_role is "not_assigned," they are redirected back to home1.php. If the user_role is "Admin," they are directed to the admin page.

```php
<?php require_once('common.php');
    session_start();

    // Ελεγχει αν ο user που ειναι στην σελιδα log_in.php εχει κανει log_in και αν εχει το σωστο SESSION
    function check_login($conn){
        // Ελεγχει αν εχει οριστει το SESSION με το user_id του user που εκανε log in
        if(isset($_SESSION['user_id']))
        {
            $id = $_SESSION['user_id'];
            $query = "select * from user where user_id = '$id' limit 1";

            $result = mysqli_query($conn , $query);
            // Ελεγχει αν υπαρχει η εγγραφη στο database
            if($result && mysqli_num_rows($result) > 0)
            {
                $user_data  = mysqli_fetch_assoc($result);
                return $user_data;
            }

        }
        // Αν δεν εχει οριστει το SESSION με το user_id εκεινου που εκανε log in τοτε σε πηγαινει παλι στο home1.php
        header("Location: home1.php");
        die;

    }

    $user_data    = check_login($conn);

    if($user_data['user_role']== "not_assigned")
    {
        header("Location: home1.php");
    }

    // Αν το user_role εκεινου που εκανε log_in ειναι ισο με "Admin" τοτε ειναι admin και τον πεταει στο admin.php
    if($user_data['user_role'] == "Admin"){
        header("Location: admin.php");
    }
```

If the user clicks on a movie, they are redirected to tickets.php, passing the movie_id in the URL.

```php
while ($row=$result->fetch_array(MYSQLI_ASSOC)){
    $movie_id = $row['movie_id'];
    $movie_name = $row['movie_name'];
    echo "<div class='movies' style='border: 3px solid orange; width: 180px; height: auto; text-align: center; background-color: lightyellow; border-radius: 6px; margin-left:5px; m
    <div class='item'><a href='tickets.php?id=$movie_id' style='text-decoration: none'><img src='images/$row[movie_name].png' width=150 height=220 style='margin-top: 10px;'><br> "

}
?>
```

There, they see the ticket purchase menu where they can select a specific screening (previously defined by the admin) and the number of tickets. An SQL statement is executed to create variables with information about the selected movie.

```php
////////// MOVIE
// pairnei apto url to id kai me vasi auto ftiaxnei kati variables pou ta xrhsimopoioume kato gia na emfanisoume alles plirofories
if (isset($_GET['id'])) {
    $movie_id = $_GET['id'];
    $sql = "SELECT * FROM movie WHERE movie_id = $movie_id";
    $result2 = $conn->query($sql);

    if ($result2->num_rows > 0) {
        $row = $result2->fetch_assoc();
        $movie_name = $row["movie_name"];
        $movie_director = $row["movie_director"];
        $movie_desc = $row['movie_desc'];

    }
}
```

The way screenings are displayed:

```php
<select id='screening_time' name='screening_id'>";

// Τα screening_time εμφανιζονται αναλογα το movie_id
$sql = "select * from screening where movie_movie_id =$movie_id ";
$result3=$conn->query($sql);
if(!$result3) die($conn->error);
while($row=$result3->fetch_array(MYSQLI_ASSOC)){
    echo "<option value='".$row['screening_id']."'>".$row['screening_time']."</option>";
}

echo "</select>
```

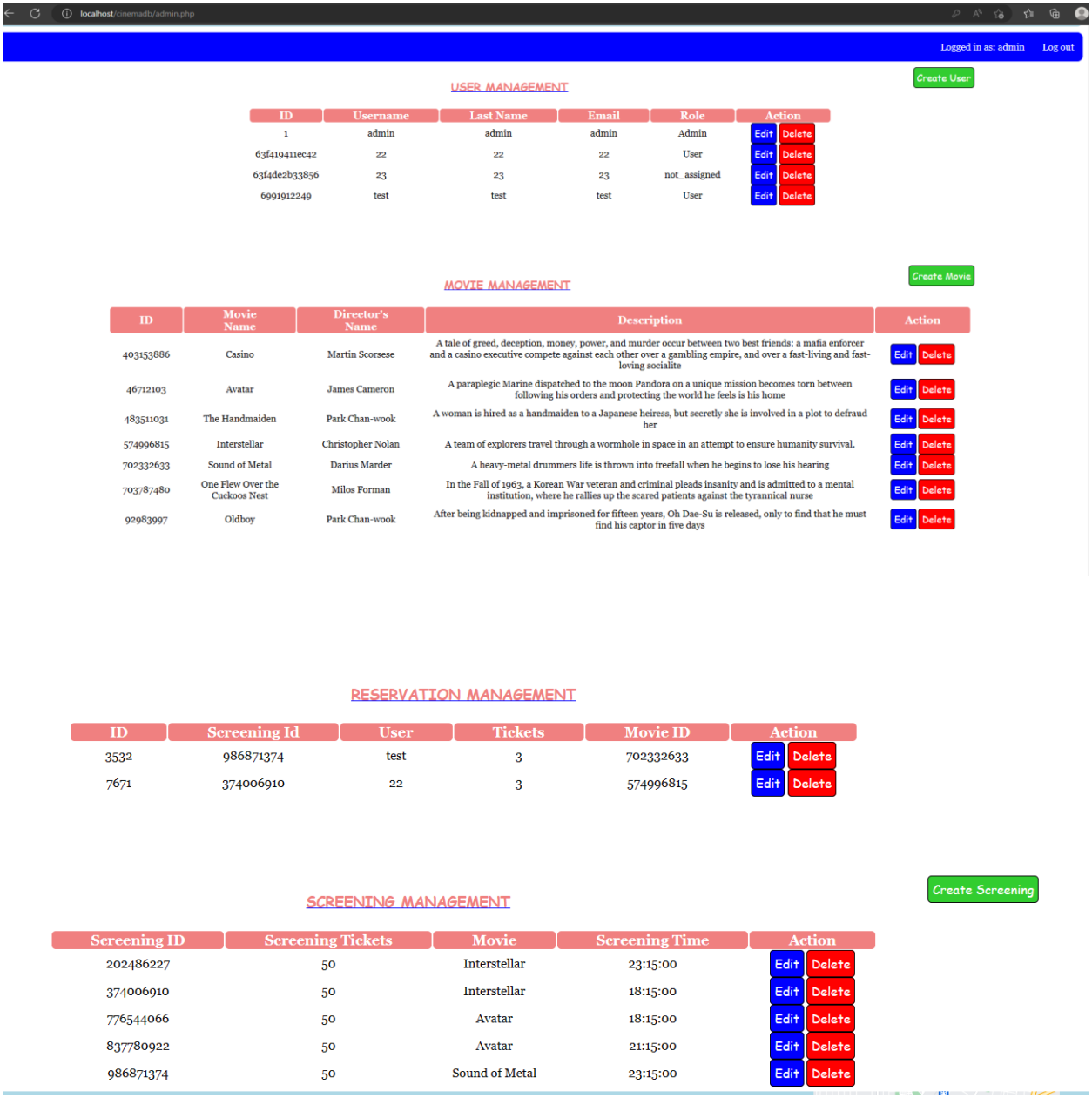
To select the number of tickets, two objects that resemble buttons have been created, along with a script that runs when one of these two "buttons" is clicked.

```
<input type='number' id='ticketAmount' name='ticketAmount' value='1'>
<a class='button' onclick='ticketPlus();'>+</a>
<a class='button' onclick='ticketMinus();'>-</a>

<script>
function ticketMinus() {
    temp = document.getElementById('ticketAmount').value;
    if(temp>1){
    document.getElementById('ticketAmount').value = temp-1;
    }
}

function ticketPlus() {
    temp = document.getElementById('ticketAmount').value;
    document.getElementById('ticketAmount').value = parseInt(temp)+1;
}
</script>
<br>
<br>
<br>

<button id='tickets_button' type='submit'>Buy tickets</button>
";
?>
<!-- Cancel Button -->
<br>
<button type="button" onclick="location.href='log_in.php'" class="cancelbutton">Cancel</button>
<?php
echo "
```

←T→	reservation_id	screening_screening_id	ticket_amount	movie_movie_id	user_user_id
☐ Επεξεργασία Αντιγραφή Διαγραφή	3532	986871374	3	702332633	6991912249

Admin.php

When an admin logs in, a user with user_role="Admin" is redirected to admin.php instead of log_in.php.



There, the admin can view everything related to the cinema system. They can also modify users, create new users, delete users, and perform the same actions for movies and screenings.

Create User

When the admin clicks on "Create User," a modal form appears (similar to when a user signs up).

```
<a class="create" onclick="document.getElementById('id01').style.display='block'" style="background-color:#32CD30; float:right;">Create User</a>
```

```
<!--USER MANAGEMENT WITH MODAL FOR CREATE USER-->
<div id="id01" class="modal">
        <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">X</span>
        <form class="modal-content" action="admin.php" method="post">
            <div class="container">
            <h1>Create User</h1>
            <hr>


           <!-- Sign UP Modal -->
           <label for="user_name"><b>Name:</b></label>
            <input type="text" placeholder="Enter your Name" name="user_name" required>

            <label for="user_surname"><b>Surname:</b></label>
            <input type="text" placeholder="Enter your Surname" name="user_surname" required>

            <label for="countries"><b>Country:</b></label>
            <select onmouseup="showCities();" class="text" name="user_country" id="countries" ></select>

            <br>
            <br>

            <label for="cities"><b>City:</b></label>
            <select class="text" name="user_city" id="cities"></select>

            <br>
            <br>

            <label for="address"><b>Address:</b></label>
            <input type="text" placeholder="Type your Address" name="user_address" required>

            <label for="user_email"><b>Email:</b></label>
            <input type="text" placeholder="Enter your Email" name="user_email" required>

            <label for="user_username"><b>Username:</b></label>
            <input type="text" placeholder="Type your Username" name="user_username" required>

            <label for="user_password"><b>Password:</b></label>
            <input type="password" placeholder="Type your Password" name="user_password" required>
            <!-- To type="password" krivei to input -->




            <div class="clearfix">
                <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancel</button>
                <button type="submit" class="signup">Create User</button>
            </div>
            </div>
```

This form has an action="admin.php," which is why the following code snippet is present in the admin.php header (similarly, we perform an insert when creating movies and screenings).

```
// o Admin κανει create καινουργιους users

if (isset($_POST['user_name'])&&isset($_POST['user_surname'])&&isset($_POST['user_country'])&&isset($_POST['user_city'])&&isset($_POST['user_address'])&&isset($_POST['user_
    $user_id= uniqid();
    $user_name=$_POST['user_name'];
    $user_surname=$_POST['user_surname'];
    $user_country=$_POST['user_country'];
    $user_city=$_POST['user_city'];
    $user_address=$_POST['user_address'];
    $user_email=$_POST['user_email'];
    $user_username=$_POST['user_username'];
    $user_password=$_POST['user_password'];


    $query="INSERT into user values"."('$user_id','$user_name', '$user_surname', '$user_country', '$user_address', '$user_email', '$user_username', '$user_password', '$user
    echo ($query);
    $result = $conn->query($query);

    if(!$query) echo ("failed").$conn->error."<br><br>";


}
```

Create Movie

```
<a class="create" onclick="document.getElementById('id03').style.display='block'">Create Movie</a>
```

```
<!--MOVIE MANAGEMENT WITH MODAL FOR CREATE MOVIE-->
<a class="create" onclick="document.getElementById('id03').style.display='block'">Create Movie</a>
<div id="id03" class="modal">
    <span onclick="document.getElementById('id03').style.display='none'" class="close" title="Close Modal">X</span>
    <form class="modal-content" action="admin.php" method="post">
        <div class="container">
        <h1>Create Movie</h1>
        <hr>

      <!-- Create Movie Modal -->
      <!-- Υπαρχει προβλημα οταν στο description παει να μπει κειμενο με αποστροφο ' μεσα καθως το μπερδευει το που τελειωνει το text
      <label for="movie_name"><b>Movie Name:</b></label>
       <input type="text" placeholder="Enter Movie's Name" name="movie_name" required>

       <label for="movie_director"><b>Movie Director:</b></label>
       <input type="text" placeholder="Enter Movie's Director" name="movie_director" required>

       <label for="movie_desc"><b>Movie Description:</b></label>
       <input type="text" placeholder="Enter Movie's Description" name="movie_desc" required>

       <br>
       <br>



       <div class="clearfix">
           <button type="button" onclick="document.getElementById('id03').style.display='none'" class="cancelbtn">Cancel</button>
           <button type="submit" class="signup">Create Movie</button>
       </div>
       </div>
    </form>
    </div>
```

```
// o Admin κανει create καινουργια movies
if (isset($_POST['movie_name'])&&isset($_POST['movie_director'])&&isset($_POST['movie_desc'])){
    $movie_id= random_num();
    $movie_name=$_POST['movie_name'];
    $movie_director=$_POST['movie_director'];
    $movie_desc=$_POST['movie_desc'];


    $query="INSERT into movie values"."('$movie_id','$movie_name', '$movie_director', '$movie_desc')";
    echo ($query);
    $result = $conn->query($query);

    if(!$query) echo ("failed").$conn->error."<br><br>";
}
```

Create Screenings

```
<a class="create" onclick="document.getElementById('id06').style.display='block'">Create Screening</a>
```

```
426         <!--Screening MANAGEMENT WITH MODAL FOR CREATE MOVIE-->
427       <a class="create" onclick="document.getElementById('id06').style.display='block'">Create Screening</a>
428       <div id="id06" class="modal">
429             <span onclick="document.getElementById('id06').style.display='none'" class="close" title="Close Modal">X</span>
430             <form class="modal-content" action="admin.php" method="post">
431                 <div class="container">
432                 <h1>Create Screening</h1>
433                 <hr>
434 
435                <!-- Create Screening Modal -->
436                 <label for="screening_tickets"><b>Screening Tickets:</b></label>
437                 <input type="text" placeholder="Screening Tickets" name="screening_tickets" required>
438 
439                 <label for="movie_id"><b>Movie Id:</b></label>
440                 <!-- Εμφανίζονται οι ταινιες σε ενα dropdown menu για να επιλεξει για πιο θα δημιουργηθει το screening -->
441                 <select id='selectMovie_id' name='movie_movie_id'>
442                     <?php
443                     $sqlMovie = "select * from movie";
444                     $resultMovies=$conn->query($sqlMovie);
445                         if(!$resultMovies) die($conn->error);
446                             while($row=$resultMovies->fetch_array(MYSQLI_ASSOC)){
447                                 echo "<option value='".$row['movie_id']."'>".$row['movie_name']."</option><br>";
448                             }
449                     ?>
450                 </select>
451 
452                 <br>
453                 <br>
454 
455                 <label for="screening_time"><b>Screening Time:</b></label>
456                 <input type="text" placeholder="Screening Time" name="screening_time" required>
457 
458                 <br>
459                 <br>
460 
461 
462 
463                 <div class="clearfix">
464                     <button type="button" onclick="document.getElementById('id06').style.display='none'" class="cancelbtn">Cancel</button>
465                     <button type="submit" class="signup">Create</button>
466                 </div>
467                 </div>
468             </form>
469         </div>
```

```
150         // o Admin κανει create καινουργια screenings
151         if (isset($_POST['screening_tickets'])&&isset($_POST['movie_movie_id'])&&isset($_POST['screening_time'])){
152             $screening_id= random_num();
153             $screening_tickets=$_POST['screening_tickets'];
154             $movie_movie_id=$_POST['movie_movie_id'];
155             $screening_time=$_POST['screening_time'];
156 
157 
158             $query="INSERT into screening values"."('$screening_id','$screening_tickets', '$movie_movie_id', '$screening_time')";
159             echo ($query);
160             $result = $conn->query($query);
161 
162             if(!$query) echo ("failed").$conn->error."<br><br>";
163         }
```

Update User

When the admin clicks the "Edit" button for a user, they are redirected to update_user.php, passing the user_id in the URL.

```
284     <td><a class='update' href='update_user.php?id=".$row['user_id']."'>Edit</a> <a class='delete' href='delete_user.php?id=".$row['user_id']."'>Delete</a></td>
285     </tr>
```

(Note: This id corresponds to the user whose button is clicked.)

```
$sqlUserManagement = "select * from user";

$resultUserManagement = $conn->query($sqlUserManagement);

    echo "
    <table class='table'>
        <thead>
            <tr>
            <th>ID</th>
            <th>Username</th>
            <th>Last Name</th>
            <th>Email</th>
            <th>Role</th>
            <th>Action</th>
        </tr>
        </thead>
        <tbody> ";
                if ($resultUserManagement->num_rows > 0) {
                    while ($row = $resultUserManagement->fetch_assoc()) {
                        echo"
                        <tr>
                        <td>".$row['user_id']."</td>
                        <td>".$row['user_username']."</td>
                        <td>".$row['user_surname']."</td>
                        <td>".$row['user_email']."</td>
                        <td>".$row['user_role']."</td>
                        <td><a class='update' href='update_user.php?id=".$row['user_id']."'>Edit</a>&nb
                        </tr>
                        ";
                    }
```

This is done because when we want to display all users in the database, we run an SQL query and use a while loop to iterate through the results.

When we are redirected to update_user.php, the fillUserFormValues() function is executed to automatically populate the values for that particular user.

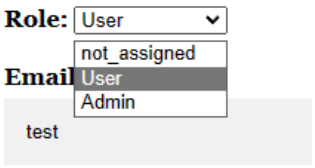
```
</head>
<body onload="fillUserFormValues()">
    <header>
        <img src="images\pngwing.com.png" width="125" height="110"><div
```

This is done by retrieving the id from the URL and running an SQL query to find the user with that id. Then, all the fields are automatically filled.

```
128                 // function που γεμιζει τα labels αναλογα με το id που παιρνει το GET
129                 echo"
130
131                 function fillUserFormValues() {
132                     ";
133
134                     $id = $_GET['id'];
135
136                     $sql2 = "select * from user having user_id ='$id'";
137                     $result4=$conn->query($sql2);
138
139
140
141                     if(!$result4) die($conn->error);
142                     while($row=$result4->fetch_array(MYSQLI_ASSOC)){
143                         echo "if(String('$id') === String('".$row['user_id']."')){
144                             document.getElementById('selectUser_username').value = String('".$row['user_username']."');
145                             document.getElementById('selectUser_id').value = String('".$row['user_id']."');
146                             document.getElementById('selectUser_role').value = String('".$row['user_role']."');
147                             document.getElementById('selectUser_email').value = String('".$row['user_email']."');
148                             document.getElementById('selectUser_name').value = String('".$row['user_name']."');
149                             document.getElementById('selectUser_surname').value = String('".$row['user_surname']."');
150                             document.getElementById('selectUser_country').value = String('".$row['user_country']."');
151                             document.getElementById('selectUser_city').value = String('".$row['user_city']."');
152                             document.getElementById('selectUser_address').value = String('".$row['user_address']."');}";
153                     }
154
155                 echo "}</script>
```

Είναι String επομενως πρεπει να υπαρχουν τα ''

The role selection for the user is done through a dropdown box.



This way, the admin can select the role for each user.

Delete User

When the admin clicks the delete button, they are redirected to delete_user.php with the id of the user whose button was clicked passed in the URL.

```
284        <td><a class='update' href='update_user.php?id=".$row['user_id']."'>Edit</a> <a class='delete' href='delete_user.php?id=".$row['user_id']."'>Delete</a></td>
```

In delete_user.php, to delete the user, reservations made by that user must be deleted first (if this is not done, errors occur because the reservation table has user_user_id as a foreign key). Additionally, when deleting a user, a check is performed to ensure that another admin is not deleted.

delete_user.php > ...

```php
<?php require_once('common.php');
    session_start();

    if (isset($_GET['id'])) {


        $id = $_GET['id'];

        $sql = "DELETE FROM reservation WHERE user_user_id = '$id' ";
        $result = $conn->query($sql);

        // cannot delete admins roles
        $sql2 = "DELETE FROM user WHERE user_id = '$id' AND user_role<>'Admin'";
        $result2 = $conn->query($sql2);

        if ($result == TRUE) {

            echo "Record deleted successfully.";

        }else{

            echo "Error:" . $sql . "<br>" . $conn->error;
        }
        header("Location: admin.php");
    }

?>
```

A similar process is followed for deleting screenings and reservations.

Update Movie

When the admin clicks the "Edit" button, they are redirected to update_movie.php with the id of the movie whose button was clicked passed in the URL.

```
358             <td><a class='update' href='update_movie.php?id=".$row['movie_id']."'>Edit</a> <a class='delete' href='delete_movie.php?id=".$row['movie_id']."'>Delete</a></td>
359             </tr>
```

Similarly to before, the details of the specific movie are automatically filled in

```
<script defer>

";
// function που γεμιζει τα labels αναλογα με το id που παιρνει το GET
echo"

    function fillMovieFormValues() {
        ";

        $id = $_GET['id'];
        $sql2 = "select * from movie";
        $result4=$conn->query($sql2);

        if(!$result4) die($conn->error);
        while($row=$result4->fetch_array(MYSQLI_ASSOC)){
            echo "if(String('$id') === String('".$row['movie_id']."')){
                document.getElementById('selectMovie_id').value = String('".$row['movie_id']."');
                document.getElementById('selectMovie_name').value = String('".$row['movie_name']."');
                document.getElementById('selectMovieDirector').value = String('".$row['movie_director']."');
                document.getElementById('selectMovie_desc').value = String('".$row['movie_desc']."');
            }";
        }

echo "};
```

Here, the action is set to update_movie.php in the same file:

```
<form class='movies' method='post' action='update_movie.php'>
```

When the admin completes the changes and clicks to save them, the code in the head section updates the corresponding record in the database. (Note: Care must be taken not to include an apostrophe ' in any input, as it may interfere with the text.)

```
50
51      // Εδω γινεται η αλλαγη στα στοιχεια του χρηστη απο τον admin
52      if (isset($_POST['movie_id'])&&isset($_POST['movie_name'])&&isset($_POST['movie_director'])&&isset($_POST['movie_desc'])){
53
54          $queryUpdateMovie="UPDATE movie SET movie_id='" . $_POST['movie_id'] . "' ,movie_name='" . $_POST['movie_name'] . "' ,movie_director='" . $_POST['movie_director'] . "' ,mov
55          echo ($queryUpdateMovie);
56          $resultUpdateMovie = $conn->query($queryUpdateMovie);
57
58          if(!$queryUpdateMovie) echo ("failed").$conn->error."<br><br>";
59          header("Location: admin.php");
60      }
61
```

```
sc'] . "' WHERE movie_id='" . $_POST['movie_id'] . "'";
```

Delete Movie

Deleting a movie follows a similar process to deleting a user.

delete_movie.php > ...

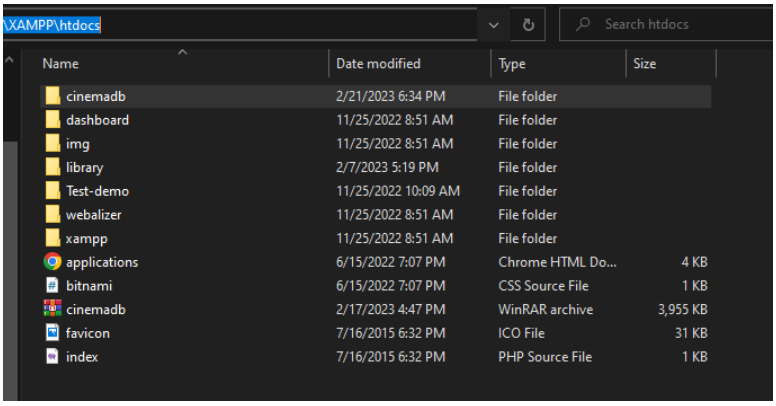
```php
<?php require_once('common.php');
    session_start();

    if (isset($_GET['id'])) {


        $id = $_GET['id'];

        //Για να γινει delete ενα movie θα πρεπει πρωτα να διαγραφουν ολα τα reservations που σχετιζονται με αυτο το movie.
        $sql = "DELETE FROM reservation WHERE movie_movie_id = '$id' ";
        $result = $conn->query($sql);


        //Για να γινει delete ενα movie θα πρεπει πρωτα να διαγραφουν ολα τα screenings που σχετιζονται με αυτο το movie.
        $sql2 = "DELETE FROM screening WHERE movie_movie_id = '$id' ";
        $result2 = $conn->query($sql2);

        $sql3 = "DELETE FROM movie WHERE movie_id = '$id' ";
        $result3 = $conn->query($sql3);
        if ($result3 == TRUE) {

            echo "Movie Record deleted successfully.";

        }else{

            echo "Error:" . $sql . "<br>" . $conn->error;
        }
        header("Location: admin.php");

    }

?>
```

The same applies to Update Reservation, Update Screening, Delete Reservation, Delete Screening, and Create Screening.

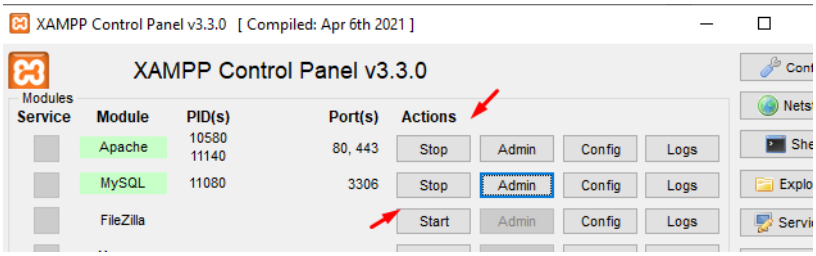
Chapter 4: Administrator Manual

The installation instructions are as follows:

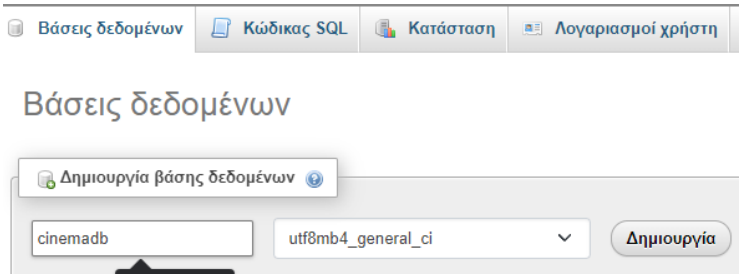
1. Take the cinemadb folder and place it in XAMPP\htdocs.



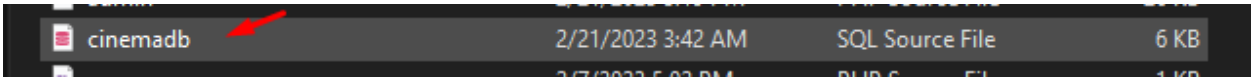
2. Open XAMPP and start Apache and MySQL.



3. Go to http://localhost/phpmyadmin/
4. Create a new database named "cinemadb".



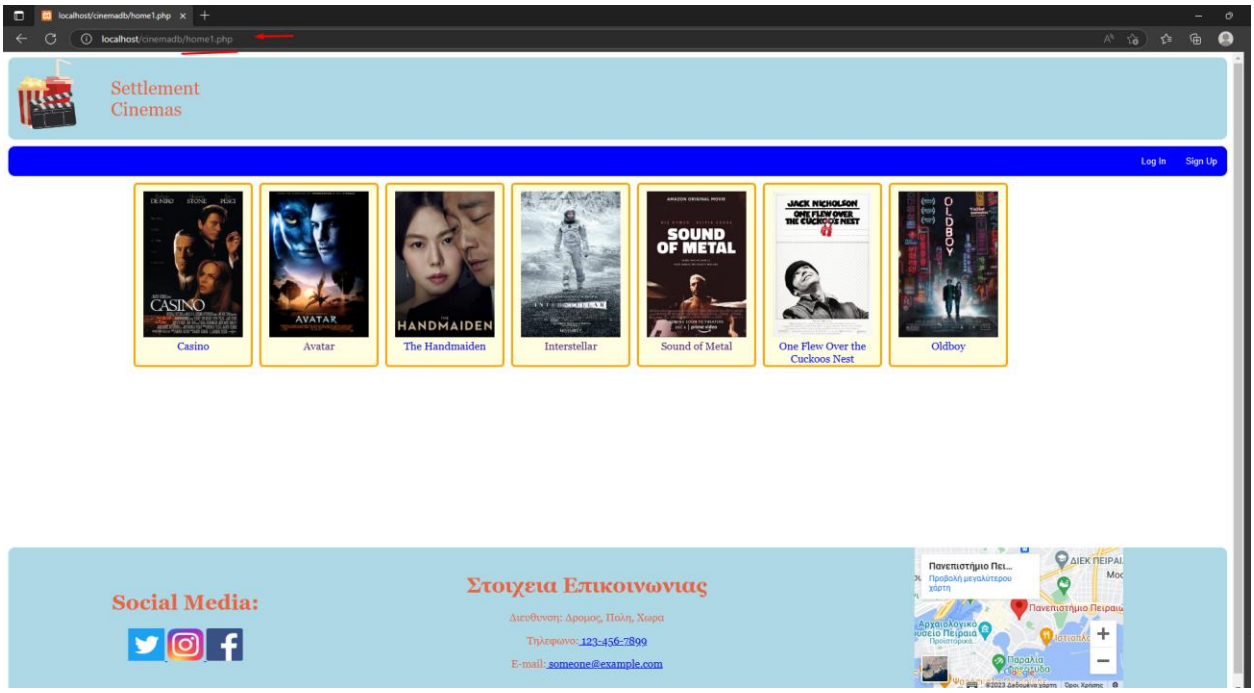
5. Go to "Import" and import the cinemadb.sql file.



6. Enter the URL localhost/cinemadb/home1.php , and you will be directed to home1.php. From here, the system can function normally.

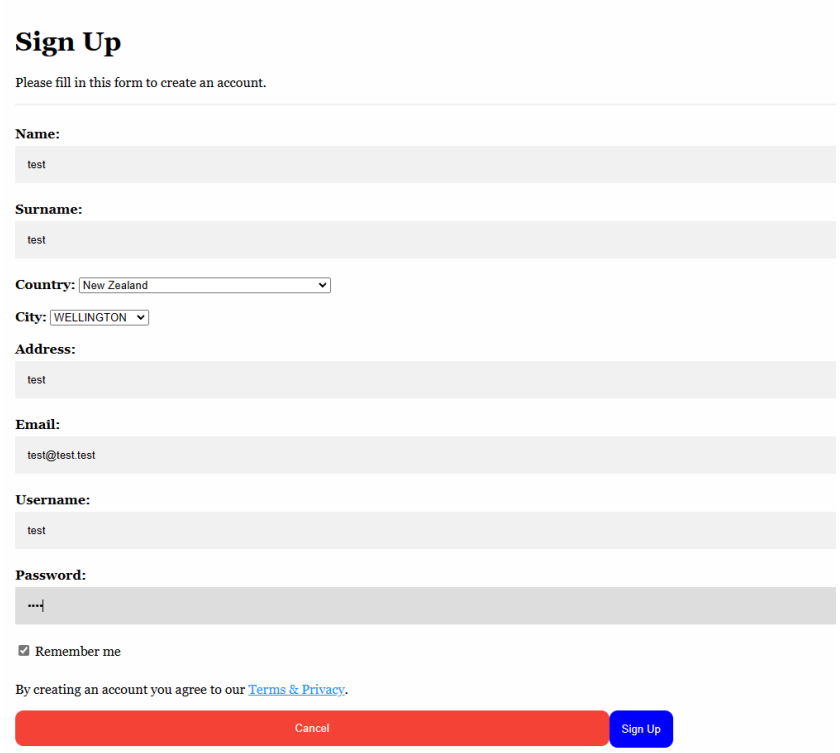
Chapter 5: User Manual

For Users: Initially, we are on home1.php.

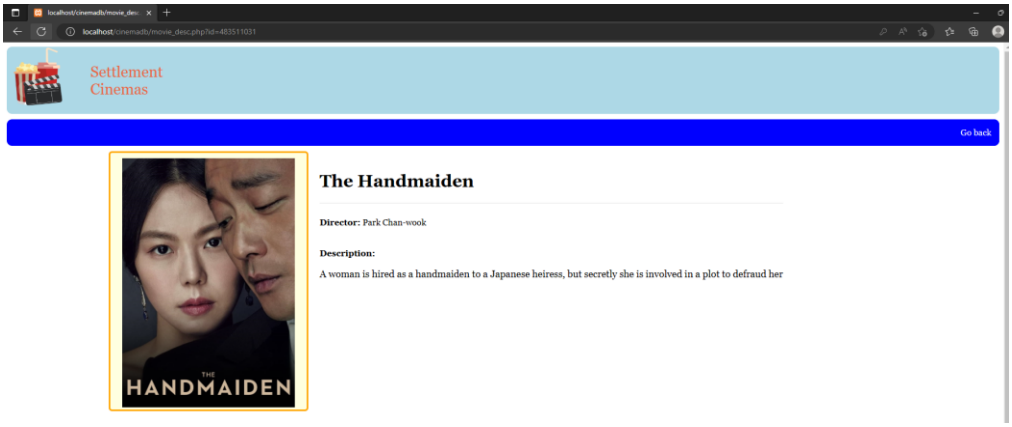


Next, a new visitor clicks the "Sign up" button.

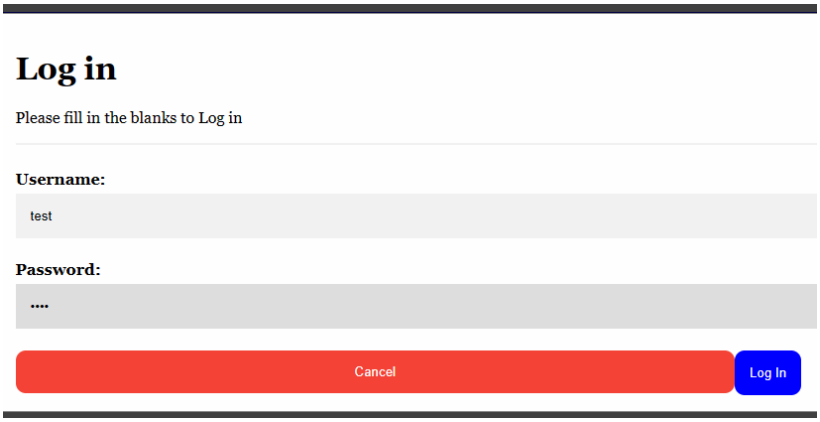
A form will appear, where the visitor fills in their details.



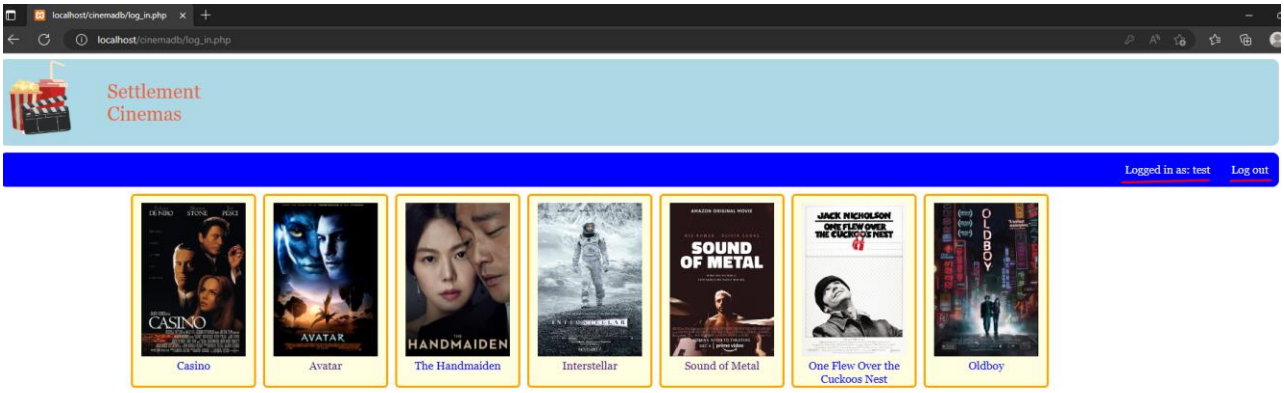
Then, they go back to home1.php and must wait for the admin to approve their request and assign them the role of a User. Until then, they can click on movies to view more information and go back to home1.php by clicking the "Go Back" button.



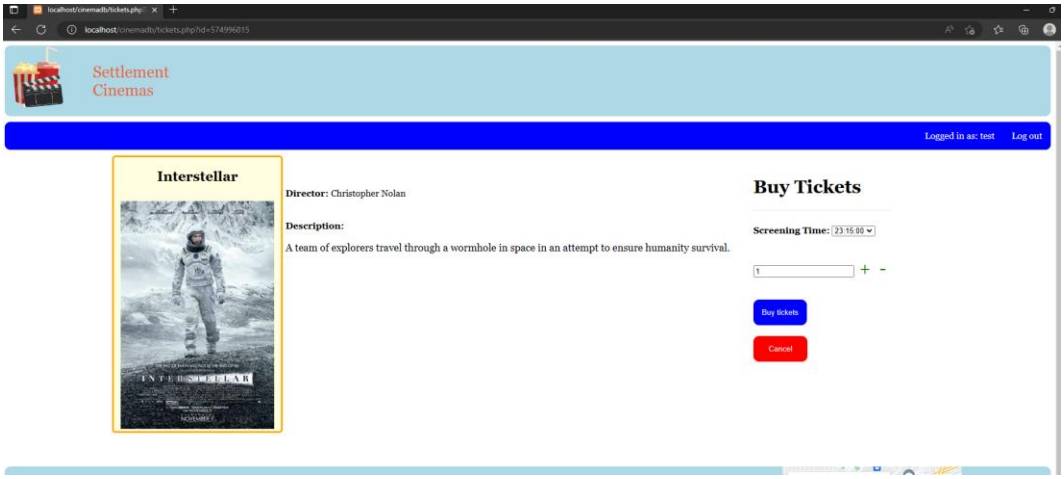
When the admin assigns the User role to the visitor, they can log in and go to log_in.php.



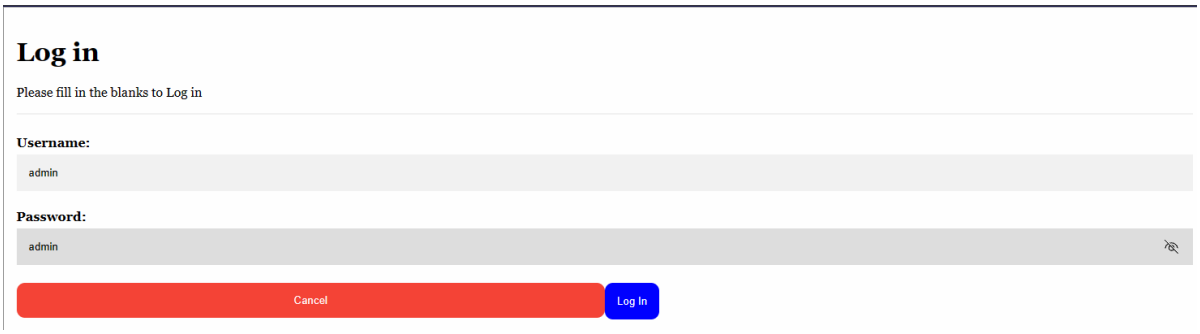
There, they can see the movies and their username displayed as logged in. There is also a "Logout" button to log out and go back to home1.php.



To purchase tickets, the user must click on a movie. This will take them to a page where they can select the number of tickets they want to buy. They also have the option to go back to the home page or log out.

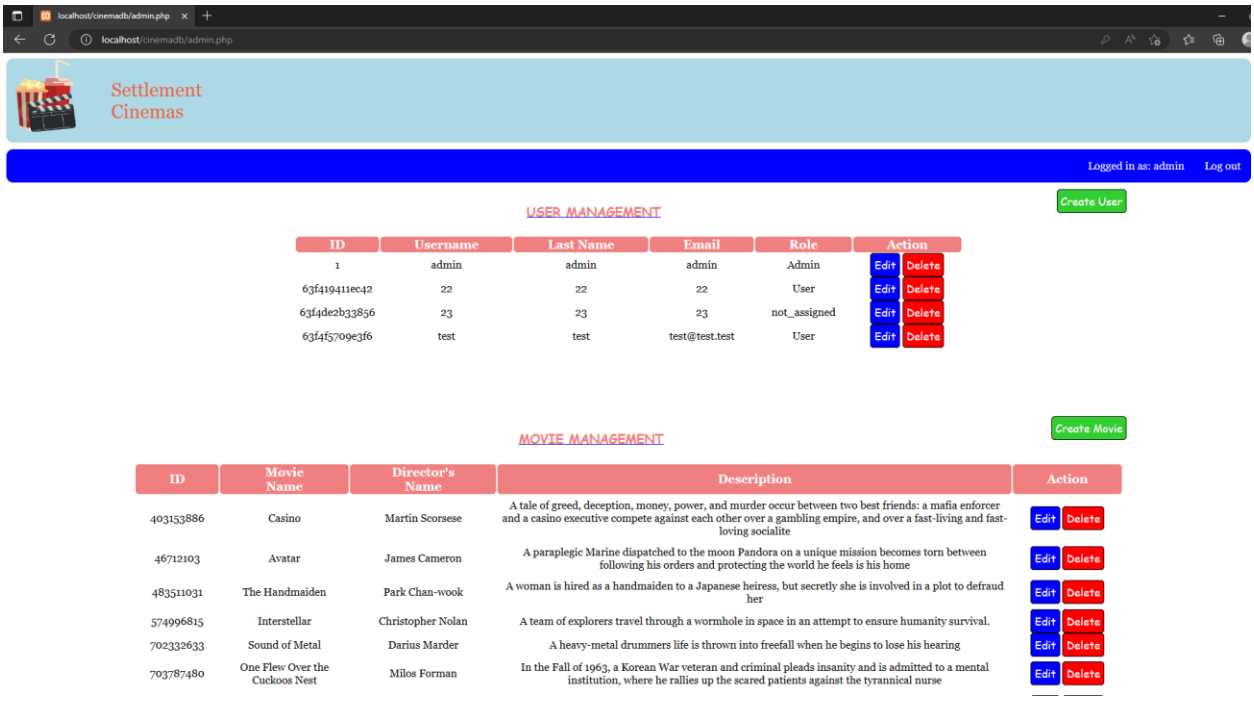


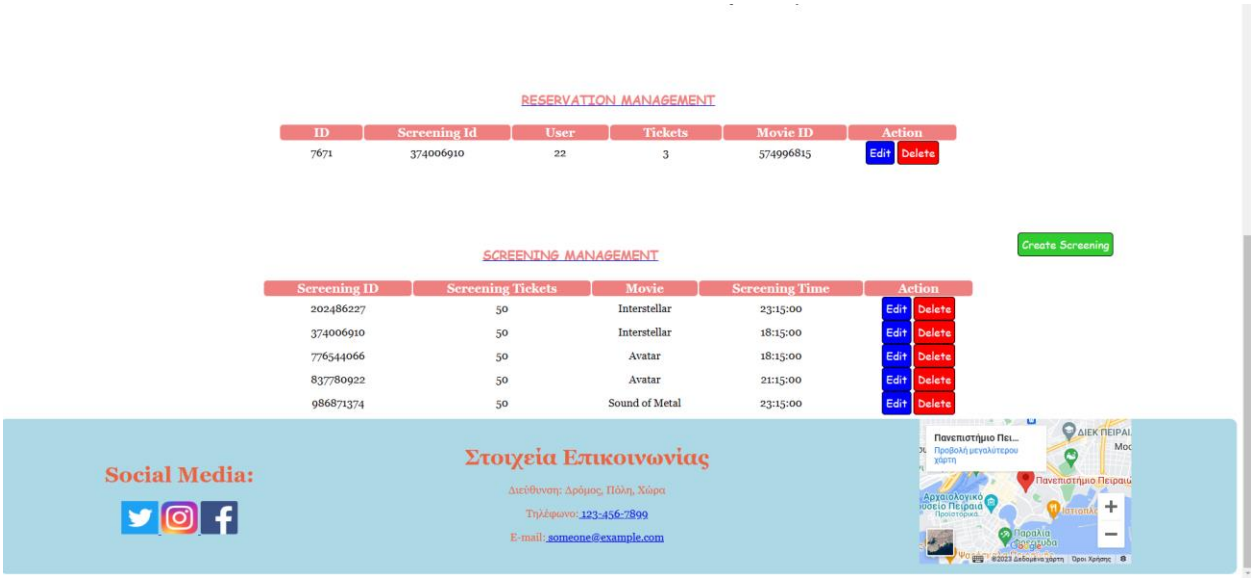
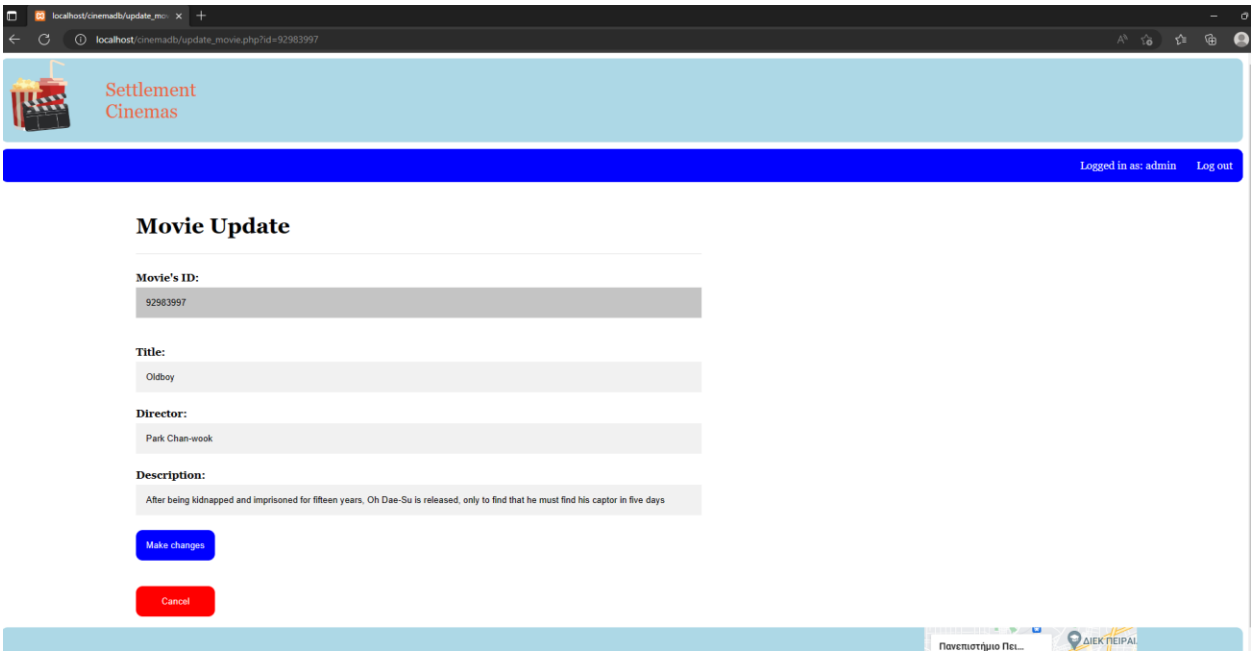
For Administrators: To access the system management page, the admin must log in with the following credentials: Username: admin, Password: admin.



These are the initial admin credentials. If additional users are assigned the role of admin, a user with the Admin role can access the system management page using their own credentials.

In Admin.php, the admin can view all system users, their roles, all movies, all reservations, and all screenings.





From this page, the admin can create new users, movies, and screenings. They can also edit any element in the database from this page by clicking the "Edit" button.