**电影评价管理系统**



课程名称：用HTML5 和 PHP编写JavaScript、jQuery 和 AJAX脚本

实验名称： 电影评价管理系统

学生姓名： 杨慕云 张煦涵

专 业： 软件工程（中外合办）

目录

[一、实验目的 3](#_Toc4298)

[二、实验环境 3](#_Toc5299)

[1.组建电影评价管理系统项目团队 3](#_Toc4629)

[2. word系列软件 3](#_Toc20299)

[6.互联网资源 3](#_Toc17802)

[三、实验内容 3](#_Toc12125)

[四、“电影评价管理系统”需求分析报告 3](#_Toc18851)

[1.1 电影评价管理系统的系统特点与介绍 3](#_Toc31557)

[(1)操作界面 3](#_Toc3565)

[(2)简便的用户模式 3](#_Toc9336)

[(3)科学的管理模式 4](#_Toc26052)

[1.2 系统功能设计 5](#_Toc459)

[1.2.1系统功能 5](#_Toc480)

[1.3功能模块设计 5](#_Toc26961)

[1.3.1、用户登录注册模块 5](#_Toc3940)

[1.3.2、电影信息管理模块 5](#_Toc3496)

[1.3.3、电影评价管理模块 6](#_Toc28752)

[1.4、系统分析 6](#_Toc25392)

[1.4.1电影评价系统结构图 6](#_Toc24700)

[2.1数据库的设计 6](#_Toc13189)

[2.1.1 用户/管理员数据表 7](#_Toc13199)

[2.1.2 评论数据表 7](#_Toc31277)

[2.1.3 电影数据表 7](#_Toc31632)

[2.2 数据库的具体信息 7](#_Toc15199)

[3.代码设计 8](#_Toc9466)

[3.2 javascript运用 9](#_Toc20038)

[3.3 php运用 14](#_Toc15485)

[3.5 ajax运用 20](#_Toc7736)

[4.代码实现 22](#_Toc29657)

[4.1界面功能介绍 22](#_Toc17071)

[4.1.1、登陆页面 22](#_Toc26225)

[4.1.2、注册界面 23](#_Toc3084)

[4.1.3、用户登录前的电影首页（看右上角的变化） 23](#_Toc25478)

[4.1.4、用户登录后的电影首页（看右上角的变化） 23](#_Toc21353)

[4.1.5、电影搜索查询 24](#_Toc29499)

[4.1.7、用户个人中心（个人评论、管理账号） 26](#_Toc3590)

[4.1.8、管理员更新电影界面 28](#_Toc13522)

[4.1.9、管理员查看管理评论界面 28](#_Toc26902)

[4.1.10、管理员管理电影信息 29](#_Toc26565)

[4.1.11、管理员管理用户 30](#_Toc6388)

[6.1程序能力 31](#_Toc16036)

[6.2缺陷和限制 31](#_Toc7948)

[6.3建议 31](#_Toc20958)

[6.4测试结论 31](#_Toc9209)

[7.组员的分工与收获 31](#_Toc31203)

[7.1组员分工 31](#_Toc12200)

[7.2组员收获 32](#_Toc21309)

**一、实验目的**

在经过一个学期的html、js、php、jquery、ajax的学习后，为了学以致用，我们做了一个电影评价管理系统。

**二、实验环境**

1.组建电影评价管理系统项目团队

2. word系列软件

3.vscode

4.phpstudy

5.phpWebstudy搭建环境

6.互联网资源

**三、实验内容**

1.需求分析，电影评价管理系统需要有什么功能。

2.搭建流程图，将功能可视化，流程化。

3.代码实现功能。

4.撰写实验报告。

**四、“电影评价管理系统”需求分析报告**

**1.1 电影评价管理系统的系统特点与介绍**

**(1)操作界面**

操作界面简洁明了，并配有适当的动画，交互性很强，能实现大部分网站的功能。

**(2)简便的用户模式**

用户系统分为3个模块：用户登录注册模块、自主评论评分模块、用户个人中心模块。

**用户登录注册：**

① 接口描述：若已有账号，则在登录表单，输入用户的用户名和密码；若无账号，则在注册表单，输入新用户的用户名和密码，并且再次确认密码。

② 实现方法：在登陆表单，接收用户名和密码，再与数据库中的数据对比，若相匹配，则进入电影评价系统；在注册表单，接收新用户的用户名和密码，若符合用户名和密码均符合要求，则将数据上传至数据库，跳转至登陆表单。

**自主评论评分：**

模块功能：可以查看电影信息，也可以给电影评分，并发表对电影的评论。

**用户个人中心：**

用户中心包括三个部分：个人中心、评论管理和账号管理。

1、个人中心：

① 模块功能：可以查看用户的基本信息。

② 实现方法：连接数据库，将用户个人信息展示。

2、评论管理：

① 模块功能：可以查看用户个人的所有评论记录，并可以在此修改或删除自己的评论。

② 实现方法：连接数据库，将评论记录展示，并设计修改和删除按钮。

**(3)科学的管理模式**

管理员系统分为4个模块：系统管理员身份认证模块、电影上传管理模块、用户评论管理模块和电影信息管理模块。

**管理员身份认证：**

① 接口描述：若已有账号，则在登录表单，输入管理员的用户名和密码，并勾选管理员复选框；若无账号，则在注册表单，输入新管理员的用户名和密码，并且再次确认密码。

② 实现方法：在登陆表单，接收用户名和密码，管理员复选框信息，再与数据库中的数据对比，若相匹配，则进入电影评价系统；在注册表单，接收新管理员的用户名和密码，管理员复选框信息，若符合用户名和密码均符合要求，则将数据上传至数据库，跳转至登陆表单。

**电影上传管理：**

① 模块功能：上传新的电影数据。

② 实现方式：管理员输入电影信息，上传到数据库。

**用户评论管理模块：**

① 模块功能：输出用户电影评论管理信息，对用户影评执行查看和删除操作。

② 过程陈述：进入管理评论界面，查看所有用户电影评论，输入影评所显示的ID，可删除相对应用户的评论。

**电影信息管理：**

① 模块功能：查看以及修改电影的基本信息。

② 实现方法：选择电影，修改信息，上传到数据库，更新数据库信息。

**1.2 系统功能设计**

**1.2.1系统功能**

用户与管理员的登录功能。

用户可以通过电影界面，查看电影信息，并对电影进行评论评分操作。

用户可以在个人中心的管理评论中对个人评论进行修改删除管理。

管理员可以对电影库里的电影进行管理，可以上传新电影。

管理员可以对电影信息进行修改，添加以及删除。

管理员可以对用户的评论进行查询以及删除操作。

**1.3功能模块设计**

**1.3.1、用户登录注册模块**

当用户登陆时，会验证用户名和密码是否对应，若对应，会根据是管理员账户或是用户账户，进入不同的界面。若账户不存在，会提示，账户不存在。若用户还未注册，可以新注册一个账号。如果用户名已存在，点击注册后会显示用户名已存在，注册失败。

**1.3.2、电影信息管理模块**

管理员对电影的简介进行修改管理。

**1.3.3、电影评价管理模块**

用户在个人中心显示评论的详细信息，并可以在这里修改删除自己的评论评分。

管理员显示所有用户评价的情况，并可以在这删除普通用户评论。

用户访问电影查询模块时显示当前查询电影的信息。

**1.4、系统分析**

**1.4.1电影评价系统结构图**



**2.数据库概述**

**2.1数据库的设计**

以整个电影评价管理系统为整体，建立一个userinformation数据库。建立三个数据表：user、comment、film。

**2.1.1 用户/管理员数据表**

需要一个user数据表，用于保存用户和管理员的用户名和密码。用户和管理信息放在同一张数据表中用manager字段值为0或1来区分用户和管理员。

**2.1.2 评论数据表**

需要一个comment数据表，用于储存用户评论的电影，以及对该电影的评论和评分。

**2.1.3 电影数据表**

需要一个film数据表，用于保存电影的各种信息。

**2.2 数据库的具体信息**

**（1）user表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 是否允许为空 | 是否主键 | 描述 |
| ID | int | 否 | 是 | 用户对应的id |
| name | text | 否 | 否 | 用户名 |
| password | text | 否 | 否 | 用户对应密码 |
| manager | text | 否 | 否 | 区分用户/管理员 |

**（2）film表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 是否允许为空 | 是否主键 | 描述 |
| ID | int | 否 | 是 | 每条评论对应的ID |
| filmname | text | 否 | 否 | 电影名 |
| editor | text | 否 | 否 | 导演名 |
| actor | text | 否 | 否 | 演员名 |
| kind | text | 否 | 否 | 电影类别 |
| language | text | 否 | 否 | 电影语言 |
| ReleaseDate | text | 否 | 否 | 上映日期 |
| time | text | 否 | 否 | 电影时长 |
| introduce | text | 否 | 否 | 电影简介 |
| image | text | 否 | 否 | 电影图片路径 |

**（3）comment表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 类型 | 是否允许为空 | 是否主键 | 描述 |
| username | text | 否 | 否 | 用户名 |
| comment | text | 否 | 否 | 用户的影评 |
| filmname | text | 否 | 否 | 电影名称 |
| ID | int | 否 | 是 | 每条评论的ID |
| score | text | 否 | 否 | 用户对电影的评分 |
| time | datatime | 否 | 否 | 发表评论的时间 |

**3.代码设计**

**3.1html运用**

在php中运用html来实现页面设计

**3.2 javascript运用**

**（1）用户/管理员的登录注册采用滑动切换界面模式：**

|  |
| --- |
| JavaScript <script>  let *flag*=true  const mySwitch=()=>{  if (*flag*) {  $(".pre-box").css("transform","translateX(100%)")  $(".pre-box").css("background-color","rgb(161,181,206)")  $("img").attr("src","img/DouBan3.png")   }else{  $(".pre-box").css("transform","translateX(0%)")  $(".pre-box").css("background-color","rgb(136,189,136)")  $("img").attr("src","img/DouBan2.png")  }  *flag*=!*flag* } </script> |

**（2）电影系统主界面设置电影轮播图来展示电影海报：**

|  |
| --- |
| JavaScript <script>  // 当前轮播图编号  let *current\_index*=-1;  // 自动轮播定时器  let *swipe\_timer*=null;  // 轮播图的图片地址与跳转链接  let *links*=[  {'image':'film/swipe1.bmp','target':'#1'},  {'image':'film/swipe2.bmp','target':'#2'},  {'image':'film/swipe3.bmp','target':'#3'},  {'image':'film/swipe4.bmp','target':'#4'},  {'image':'film/swipe5.bmp','target':'#5'},  ];   // 需要操作到的元素  let *swipe*=*document*.getElementById('swipe');  let *swipe\_bg*=*document*.getElementById('swipe\_bg');  let *swipe\_img\_box*=*document*.getElementById('swipe\_img\_box');  let *swipe\_link*=*document*.getElementById('swipe\_link');  let *swipe\_img*=*document*.getElementById('swipe\_img');  let *swipe\_select*=*document*.getElementById('swipe\_select');  let *swipe\_btn\_left*=*document*.getElementById('swipe\_btn\_left');  let *swipe\_btn\_right*=*document*.getElementById('swipe\_btn\_right');   // 事件  // 切换图片  let select=(index)=>{  // 停止播放  stop();  // 转数字  index=Number(index);  // 越界超过最大数量,直接返回  if(index>=*links*.length){  return;  }  // 选中当前已选中的,直接返回  if(*current\_index*==index){  return;  }  // 取消当前指示点的选中状态  if(*current\_index*>-1){  *swipe\_select*.children[*current\_index*].classList.remove('checked');  }  // 变更当前轮播图的编号  *current\_index*=index;  // 找到当前元素  let current\_link=*links*[*current\_index*];  // 背景变化  *swipe\_bg*.style.backgroundImage='url('+current\_link.image+')';  // 前景变化  *swipe\_img*.setAttribute('src',current\_link.image);  // 链接变化  *swipe\_link*.setAttribute('href',current\_link.target);  // 增加新的指示点的选中状态  *swipe\_select*.children[*current\_index*].classList.add('checked');  };  // 自动切换图片  let autoSelect=(index)=>{  // 转数字  index=Number(index);  // 越界超过最大数量，直接返回  if(index>=*links*.length){  return;  }  // 选中当前已选中的，直接返回  if(*current\_index*==index){  return;  }  // 取消当前指示点的选中状态  *swipe\_select*.children[*current\_index*].classList.remove('checked');  // 变更当前轮播图的编号  *current\_index*=index;  // 找到当前元素  let current\_link=*links*[*current\_index*];  // 前景图片  // 第一步调整过渡时间  *swipe\_img*.style.transition='opacity 0.5s ease-in 0s';  // 第二步调整不透明度为0.2  *swipe\_img*.style.opacity=0.2;  // 第三步延迟变换img图片，并重新定义透明度以及过渡时间和过渡方式  setTimeout(() => {  // 背景变化  *swipe\_bg*.style.backgroundImage='url('+current\_link.image+')';  // 前景变化  *swipe\_img*.setAttribute('src',current\_link.image);  // 链接变化  *swipe\_link*.setAttribute('href',current\_link.target);  // 不透明度变化  *swipe\_img*.style.transition='opacity 0.7s ease-out 0s';  *swipe\_img*.style.opacity=1;  // 增加新的指示点选中状态  // 如果已经通过手动点击了，选中则此处不再执行  if(!*document*.querySelector('.swipe .checked')){  *swipe\_select*.children[*current\_index*].style.transition='background-color 0.5s';  *swipe\_select*.children[*current\_index*].classList.add('checked');  }  }, 500);  };  // 播放  let play=()=>{  // 3秒切换一次  *swipe\_timer*=setInterval(()=>{  // 设置新的index  let index=*current\_index*+1;  // 右翻越界，切到第一张  if(index>=*links*.length){  index=0;  }  // 加载新图片（这里选择自动，增加切换效果）  autoSelect(index);  },3000);  };  // 停止  let stop=()=>{  if(*swipe\_timer*){  clearInterval(*swipe\_timer*);  *swipe\_timer*=null;  }  };  // 初始化  let init=()=>{  for(let i=0;i<*links*.length;i++){  // 创建a元素  let item=*document*.createElement('a');  // 修改属性  item.setAttribute('class','item');  item.setAttribute('href','#');  item.setAttribute('data-index',i);  // 追加元素  *swipe\_select*.appendChild(item);  }  // 默认第一张  select(0);  // 绑定各个事件并开始轮播  bind();  play();  };  // 绑定  let bind=()=>{  // 左翻事件监听  *swipe\_btn\_left*.addEventListener('click',()=>{  // 设置新的index  let index=*current\_index*-1;  // 左翻越界，切到最后一张  if(index<0){  index=*links*.length-1;  }  // 加载新图片  select(index);  });  // 右翻事件监听  *swipe\_btn\_right*.addEventListener('click',()=>{  // 设置新的index  let index=*current\_index*+1;  // 右翻越界，切到第一张  if(index>=*links*.length){  index=0;  }  // 加载新图片  select(index);  });  // 循环绑定指示器点击事件  for(const key in *swipe\_select*.children){  if(*swipe\_select*.children.hasOwnProperty(key)){  const element=*swipe\_select*.children[key];  element.addEventListener('click',(e)=>{  // 取消默认点击跳转  e.preventDefault();  // 跳转到当前指示点中data-index所指定的图片  select(e.target.dataset.index);  });  }  }  // 绑定鼠标移入事件  *swipe*.addEventListener('mouseover',(e)=>{  // 防止鼠标从子元素移出时触发  if(e.relatedTarget&&*swipe*.compareDocumentPosition(e.relatedTarget)==10){  stop();  }  });  // 绑定鼠标移出事件  *swipe*.addEventListener('mouseout',(e)=>{  // 防止鼠标从子元素移出时触发  if(e.relatedTarget&&*swipe*.compareDocumentPosition(e.relatedTarget)==10){  play();  }  });  // 绑定鼠标移动事件  *swipe*.addEventListener('mousemove',(e)=>{  stop();  });  };   // 页面加载完毕，执行初始化  *window*.addEventListener('load',()=>{  init();  }) </script> |

**(3)用户评分系统（星星显示）：**

|  |
| --- |
| JavaScript <script> document.querySelectorAll('.user-rating').forEach(userRating => {  const rating = userRating.getAttribute('data-rating'); // 获取评分  fillStars(userRating, rating); }); </script> |

**3.3 php运用**

代码基本由php文件实现，php脚本能够很好地实现数据的交互以及页面的展示，通过php技术我们能够很好地展现我们的想法。

（1）**注册登录界面的php连接数据库代码：**

|  |
| --- |
| PHP <?php   session\_start(); // 开启会话  $conn = new mysqli('localhost', 'root', 'yang2004020015', 'userinformation');  if ($conn->connect\_error) {  die("数据库连接失败:" . $conn->connect\_error);  }  $conn->set\_charset("utf8");  if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['register'])) {  $username = $\_POST["username"] ?? "";  $password = $\_POST["password"] ?? "";  $passwordVerified = $\_POST["password\_verified"] ?? "";  $manager = isset($\_POST['manager']) ? 1 : 0;  if (empty($\_POST["username"]) || empty($\_POST["password"])) {  die("名称和密码不能为空");  }    // 查询数据库中是否已存在相同的用户名  $query = "SELECT name FROM user WHERE name = ?";  $stmt = $conn->prepare($query);  $stmt->bind\_param("s", $username);  $stmt->execute();  $stmt->store\_result();  if ($stmt->num\_rows > 0) {  echo '<script>alert("用户名已存在");</script>';  } else {  $stmt = $conn->prepare("INSERT INTO user (name, password, manager) VALUES (?, ?, ?)");  $stmt->bind\_param("sss", $username, $password, $manager);  if ($stmt->execute()) {  echo '<script>alert("注册成功");</script>';  } else {  echo "注册失败，请稍后重试";  }    $stmt->close();  }  }  if($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST['login'])){  $username = $\_POST["username"] ?? "";  $password = $\_POST["password"] ?? "";  $manager = isset($\_POST["manager"]) ? $\_POST["manager"] : 0;  $stmt = $conn->prepare("SELECT name, password, manager FROM user WHERE name = ?");  if ($stmt === false) {  die('查询准备阶段出错: '.$conn->error);  }  $stmt->bind\_param("s", $username);  $stmt->execute();  $result = $stmt->get\_result();  if ($result->num\_rows === 1) {  $row = $result->fetch\_assoc();  if ($password==$row['password']) {  $\_SESSION['username'] = $username; // 设置用户名的 Session 变量    if ($manager && $row['manager']) {  $\_SESSION['manager'] = true;   setcookie("username",$username, time()+3600, "/");  setcookie("type",$manager, time()+3600, "/");//设置cookie  header("Location: homepage\_manager\_all.php");    exit();  } else if (!$manager && !$row['manager']) {  $\_SESSION['manager'] = false; // 设置普通用户权限的 Session 变量  echo "普通用户登录成功";  setcookie("username", $username, time()+3600, "/");  setcookie("type", $manager, time()+3600, "/");  header("Location: homepage\_user.php");  exit();  } else {  echo '<script>alert("用户权限错误");</script>';  }  } else {  echo '<script>alert("用户名密码错误");</script>';  }  } else {  echo '<script>alert("用户不存在");</script>';  }  ob\_end\_flush();  } ?> |

（2）**更改用户名和密码界面的php代码：**

|  |
| --- |
| PHP <?php ob\_start(); // 开启输出缓冲 $conn = mysqli\_connect("localhost", "root", "yang2004020015", "userinformation"); if ($conn->connect\_error) {  die("连接数据库失败: " . $conn->connect\_error); } $updateMessage = ''; if ($\_SERVER["REQUEST\_METHOD"] == "POST" ) { if(isset($\_COOKIE["username"])){  if(isset($\_POST['newUsername'])){  $newUsername = $\_POST['newUsername'];  $currentUsername = $\_COOKIE["username"];   // 检查新用户名是否已经存在  $stmt = $conn->prepare("SELECT \* FROM user WHERE name = ?");  $stmt->bind\_param("s", $newUsername);  $stmt->execute();  $result = $stmt->get\_result();   if ($result->num\_rows > 0) {  // 用户名已存在  echo '<script>alert("用户名已存在,更新不成功");</script>';  $updateMessage = "用户名已存在，无法更新";  } else {  // 用户名不存在，可以更新  $stmt = $conn->prepare("UPDATE user SET name = ? WHERE name = ?");  $stmt->bind\_param("ss", $newUsername, $currentUsername);  $stmt->execute();   if ($stmt->affected\_rows > 0) {  setcookie("username", $newUsername, time() + (86400 \* 30), "/");  $updateMessage = "用户名修改成功";  echo '<script>alert("用户名更新成功");</script>';  header("Location: ".$\_SERVER['PHP\_SELF']);//刷新页面  exit;  } else {  $updateMessage = "更新失败: " . $conn->error;  }  $stmt->close();  }  }   if(isset($\_POST['newPassword'])){  // 修改密码的逻辑  $newPassword = isset($\_POST['newPassword']) ? $\_POST['newPassword'] : '';  // 获取当前登录的用户名  $currentUsername = $\_COOKIE["username"];  $updatePasswordSql = "UPDATE user SET password='$newPassword' WHERE name='$currentUsername'";  if ($conn->query($updatePasswordSql) === TRUE) {  echo '<script>alert("密码修改成功");</script>';  } else {  echo "更新密码失败: " . $conn->error;  }  }    } else {  $error\_message = "未登录，无法修改用户名或密码";  } } $conn->close(); ob\_end\_flush(); // 结束并发送输出缓冲 ?> |

（3）**搜索电影php：**

|  |
| --- |
| PHP <?php $conn = mysqli\_connect("localhost", "root", "yang2004020015", "userinformation");  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error); }  if (isset($\_GET['search'])) {  $search = mysqli\_real\_escape\_string($conn, $\_GET['search']);  $sql = "SELECT filmname FROM film WHERE filmname LIKE '%$search%' LIMIT 10";  $result = mysqli\_query($conn, $sql);   if (mysqli\_num\_rows($result) > 0) {  while($row = mysqli\_fetch\_assoc($result)) {  echo "<p><a href='movie\_introduce.php?filmname=" . urlencode($row["filmname"]) . "'>" . htmlspecialchars($row["filmname"]) . "</a></p>";  }  } else {  echo "<p>No results found</p>";  } }  mysqli\_close($conn); ?>   <?php  $conn = mysqli\_connect("localhost", "root", "yang2004020015", "userinformation");  if ($conn->connect\_error) {  die("Connection failed: " . $conn->connect\_error);  }   if (isset($\_GET['search'])) {  $search = mysqli\_real\_escape\_string($conn, $\_GET['search']);  $sql = "SELECT filmname, image FROM film WHERE filmname LIKE '%$search%'"; //搜索与输入相似的电影名  $result = mysqli\_query($conn, $sql);  $num\_results = mysqli\_num\_rows($result);  if ($num\_results > 0) {  if ($num\_results == 1) {  $row = mysqli\_fetch\_assoc($result);  header("Location: movie\_introduce.php?filmname=" . urlencode($row["filmname"]));  exit;  }   } else {    }  }   mysqli\_close($conn); ?> |

（4）**退出登录php：**

|  |
| --- |
| PHP <?php    setcookie("username", "", time() - 3600, "/");  // 清除其他cookie  setcookie("type", "", time() - 3600, "/");  echo "退出登录";  header("Location:http://localhost/Login2.php");  exit;// 清除完毕，结束脚本   ?> |

**3.5 ajax运用**

**（1）搜索电影时出现的下拉框：**

|  |
| --- |
| JavaScript <script>  $(document).ready(function() {  $("#searchBox").on("input", function() {  var query = $(this).val();  console.log("Query: ", query);   if(query.length > 0) {  $.ajax({  url: "search\_suggestions.php",  method: "GET",  data: { search: query },  success: function(data) {  console.log("Data received: ", data);   $("#suggestionBox").html(data);  $("#suggestionBox").show();  }  });  } else {  $("#suggestionBox").hide();  }  });  }); </script> |

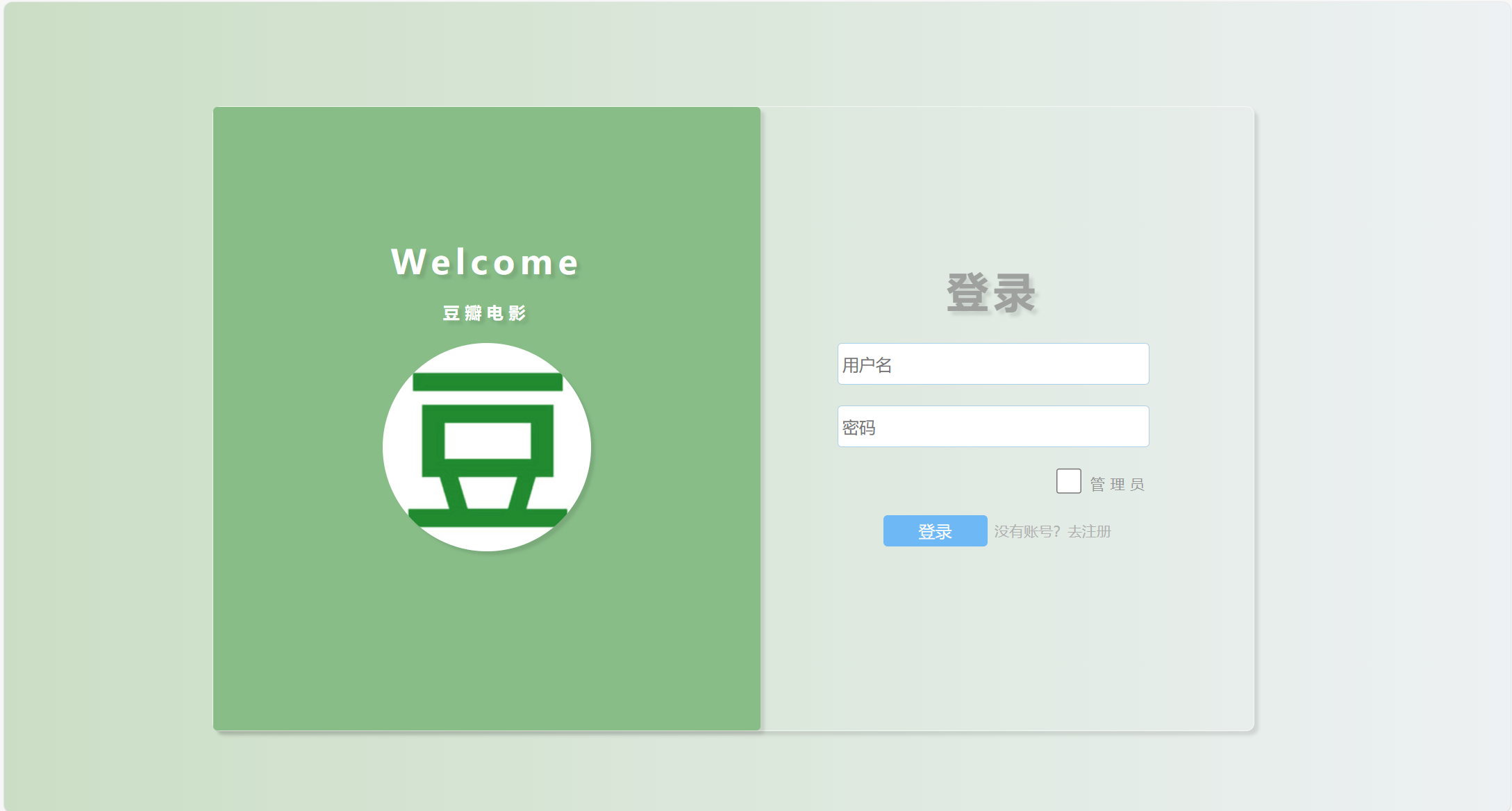
**（2）更改用户名时输入时会判断用户名是否存在：**

|  |
| --- |
| JavaScript  <script>  $(document).ready(function() {  // 当输入框发生变化时触发事件  $("#username").on("input", function() {  var newUsername = $(this).val(); // 获取用户输入的新用户名  checkUsername(newUsername); // 调用函数检查用户名是否已存在  });   // 检查用户名是否已存在  function checkUsername(username) {  $.ajax({  url: "checkUsername.php",  type: "POST",  data: {  username: username  },  success: function(response) {  if (response === "exist") {  if (username === "<?php echo $\_COOKIE['username']; ?>") {  // 当前登录用户的用户名已存在，则不允许修改  $("#username-error").text("用户名已存在");  $("#submitButton").prop("disabled", true);  } else {  // 其他用户的用户名已存在  $("#username-error").empty();  $("#submitButton").prop("disabled", false);  }  } else {  // 用户名可用  $("#username-error").empty();  $("#submitButton").prop("disabled", false);  }  },  error: function() {  alert("请求失败");  }  });  }  }); </script> |

**4.代码实现**

**4.1界面功能介绍**

**4.1.1、登陆页面**

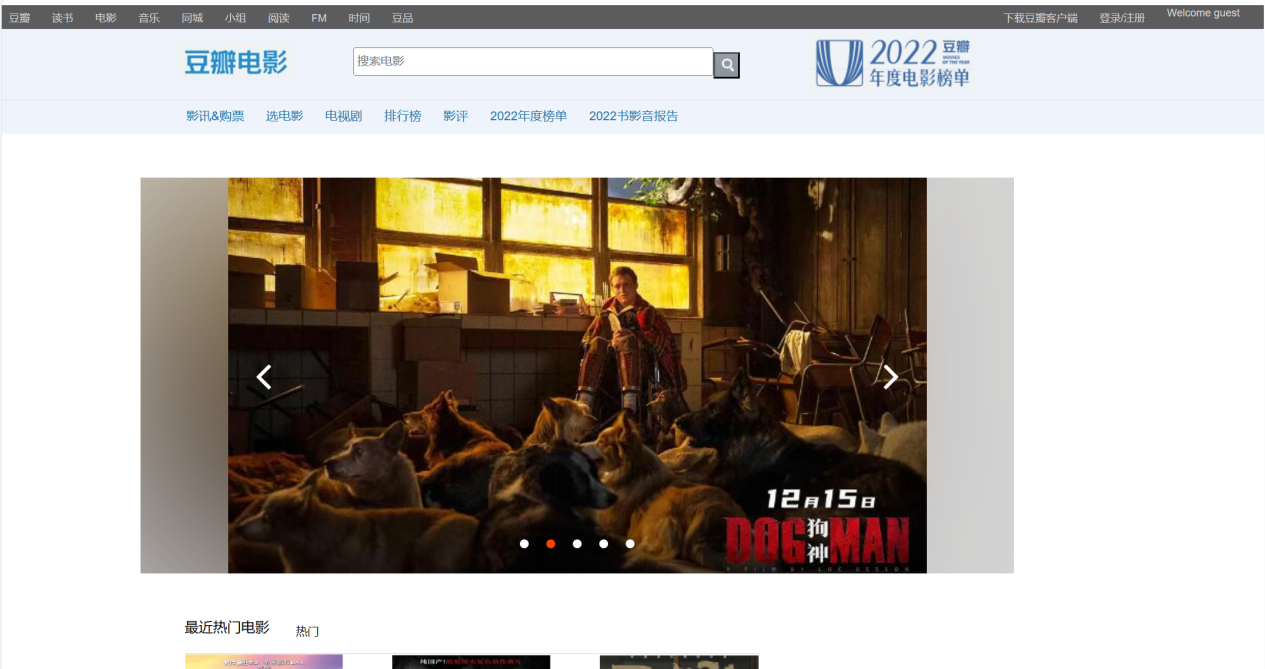


**4.1.2、注册界面**



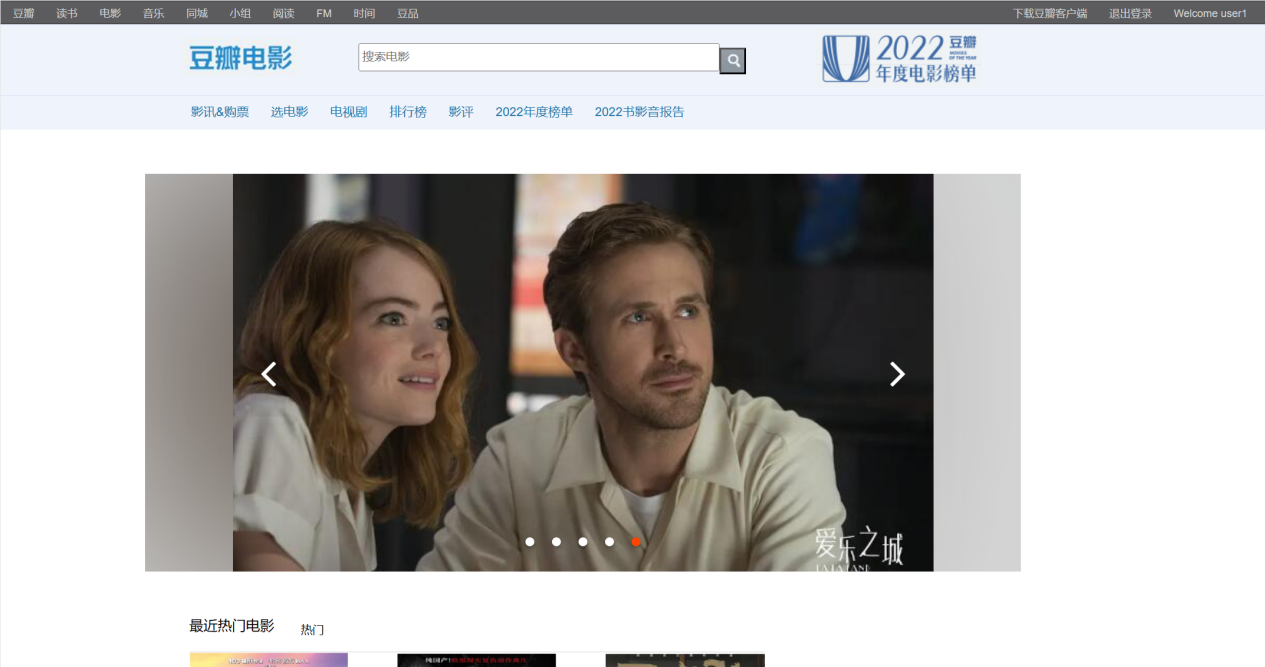
**4.1.3、用户登录前的电影首页**

在登录之前，电影首页右上方显示的是welcome guest（欢迎游客）



**4.1.4、用户登录后的电影首页**

在用户登录后，界面右上方会显示welcome 用户名（欢迎某某用户），用户信息将从数据库调取。

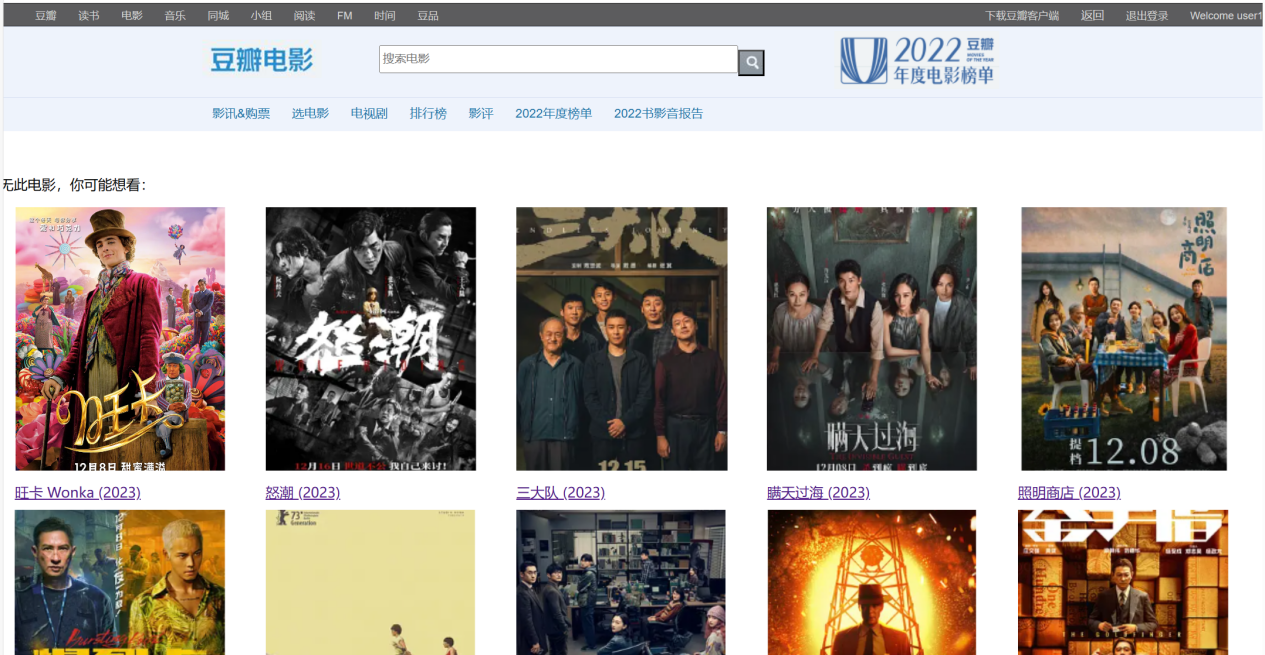




**4.1.5、电影搜索查询**

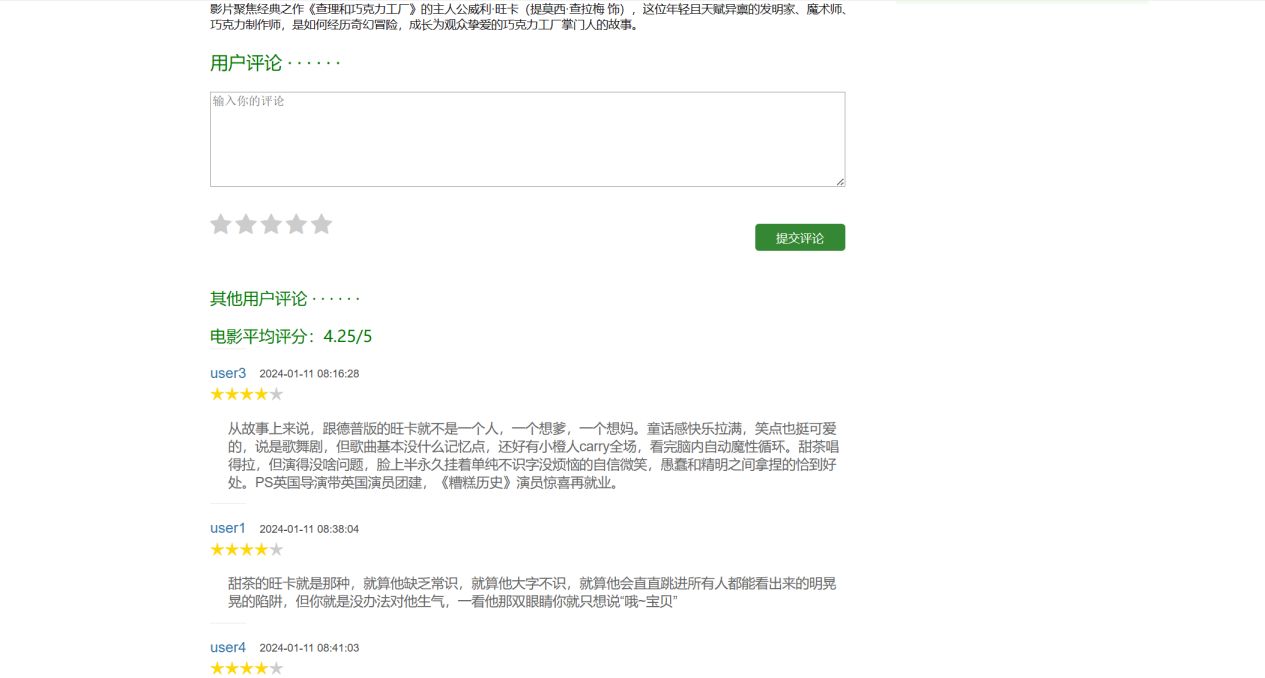


如果没有该电影：



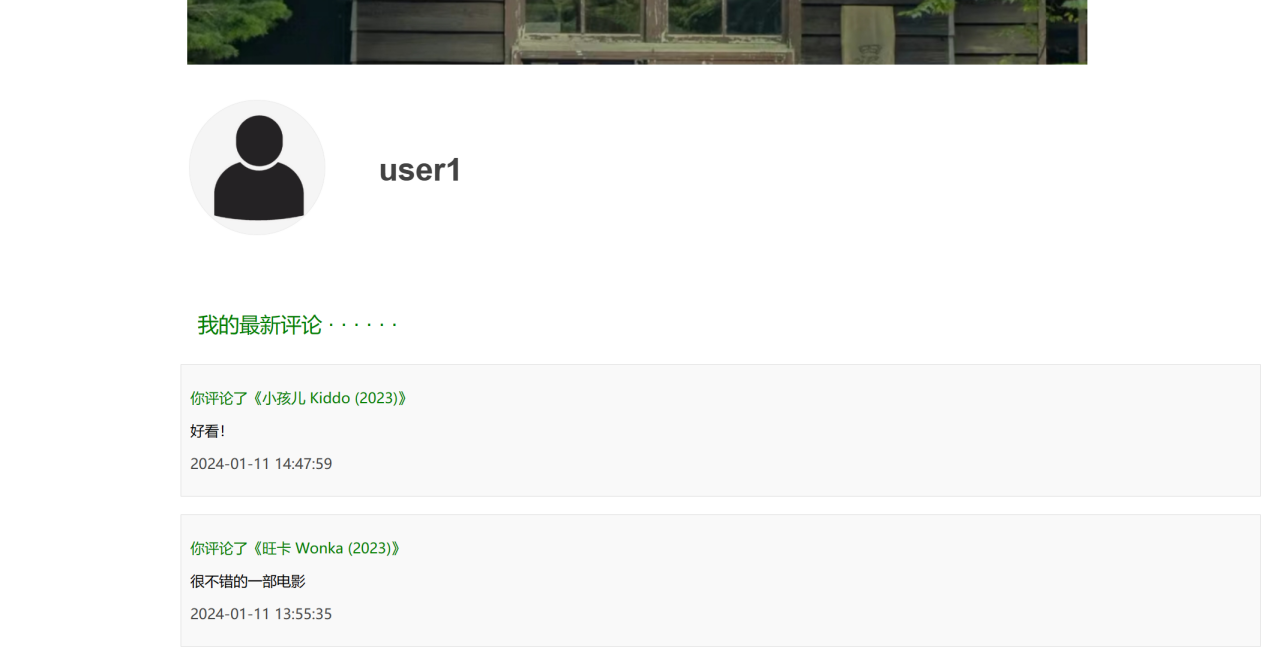
**4.1.** **6、电影信息界面（用户评论评分）**



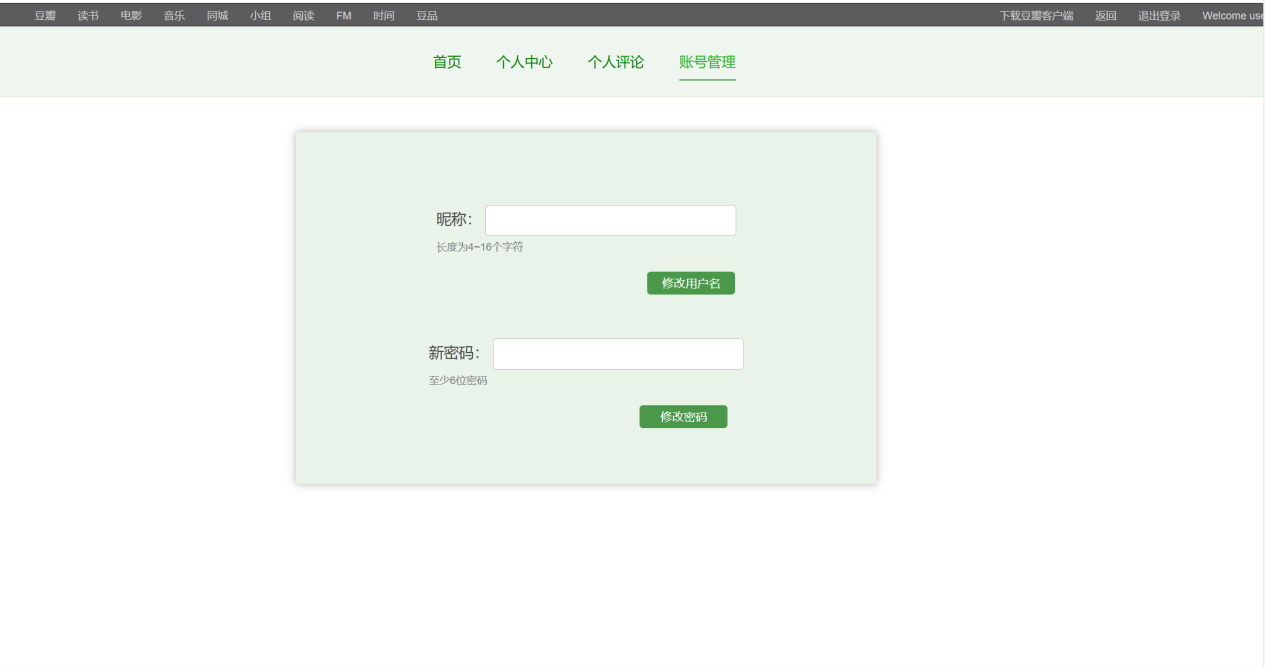


**4.1.7、用户个人中心（个人评论、管理账号）**









**4.1.8、管理员更新电影界面**





**4.1.9、管理员查看管理评论界面**





**4.1.10、管理员管理电影信息**





**4.1.11、管理员管理用户**



**5、资料的查询**

在界面设计方面，我们主要参考仿照了豆瓣电影网站的界面设计，在此主色调基础之上，添加改编，我们自己的元素风格。同时我们还通过Font Awesome 这个强大且灵活的图标库，在我们的界面中引入一些漂亮的图标，提升用户界面的质量和吸引力。

面对困难，我们常常采取阅读CSDN博客、GitHub、B站等网站上的优秀代码的方式来学习。通过模仿他人的代码并加以学以致用，我们能够更好地理解和应用这些知识到我们的程序中。这种方式不仅使我们获得了宝贵的经验，还为我们的项目开发带来了巨大的进步。不断润色修改这些代码，我们能够提升代码的质量和效率，使之更符合我们的需求。总之，借鉴他人的代码并运用到自己的程序中，是一种非常有益且富有成效的学习方法。

**6. 评价**

**6.1程序能力**

针对实验要求，能够基本完成大作业的要求，实现电影评价管理系统的基本功能。有着较强的人机交互。

**6.2缺陷和限制**

1. 部分的注意弹窗没有实现。
2. 管理员管理评论时不能根据用户或者 电影来查看评论。

**6.3建议**

尽量实现其他功能，完善电影评价管理系统的功能。

**6.4测试结论**

针对实现了电影评价管理系统的功能模块,在测试中并未发现有错误，但缺少了某些功能，在接下来的学习中我们会尽量完善它。

**7.组员的分工与收获**

**7.1组员分工**

**组员：杨慕云**

职责：框架搭建，数据库连接和功能实现，编写报告

**组员：张煦涵**

职责：搜集资料，动态界面设计，调整框架细节，编写报告

|  |  |  |
| --- | --- | --- |
| 姓名 | 身份 | 占比 |
| 杨慕云 | 组员 | 50% |
| 张煦涵 | 组员 | 50% |

**7.2组员收获**

**张煦涵：**在本次大作业中，我巩固了本学期学习的 HTML、CSS、JavaScript、jQuery、Ajax、PHP 和 MySQL 等技术知识。通过大作业的实践，我成功地运用了所学技术，也不断拓展了自己的知识库，学习到了很多课外的东西，例如一些强大的工具让界面变得更让美观灵活。其中，我主要负责的是前端界面设计工作，因此更加深入的了解掌握了如何对网页进行样式美化，包括布局设计、颜色搭配、字体排版以及响应式设计等。掌握了盒模型、定位、浮动等概念，以及Flexbox和Grid等布局的运用。此外掌握了一些交互式特效和动态功能，比如制作轮播图、响应用户操作、验证表单输入等。同时，在完成作业的过程中，查阅网上资料的经历也让我感受到了这门课程的多样性和包容性。最重要的是，和杨慕云同学两个人的团队合作中，我们分工明确，每个人都清楚自己的角色和责任，提高了整个大作业的效率，并且我们通过良好的沟通，及时分享信息、讨论问题，解决了不少困难，不断优化程序。

**杨慕云：** 在本次课设中，我收获了许多关于 PHP 数据库连接和功能实现的知识，以及使用 AJAX 实现异步交互式功能和设置 Cookie，还有数据库的读取操作。1.数据库连接：学会了使用 PHP 连接数据库的方法，包括连接 MySQL 数据库和设置连接参数。2.数据库功能实现：学会了使用 PHP 实现各种数据库功能，如插入数据、更新数据、删除数据和查询数据和读取数据。3.异步交互式功能，通过学习 AJAX，我能够实现异步交互式功能。4.Cookie 设置：我学会了使用 PHP 设置和读取Cookie。通过设置 Cookie，我可以在用户浏览器上存储当前登录用户，并在后续的请求中使用。这些知识和技能为我未来的 web 开发道路打下了坚实的基础。我相信这些收获将成为我在实际项目中运用的有力工具，提升我的能力和效率。我会继续努力学习和实践，进一步掌握这些技术，并在未来的项目中应用它们。