《WEB项目课程设计》实训结题报告

（2017~2018 学年第 春季学期）

班级：15软工A1班　　　　姓名：褚旭

编号：　　　　　　　　　学号：20154830247

结构复杂，请使用目录功能！

一、系统（或网站）功能简介

1. 题目

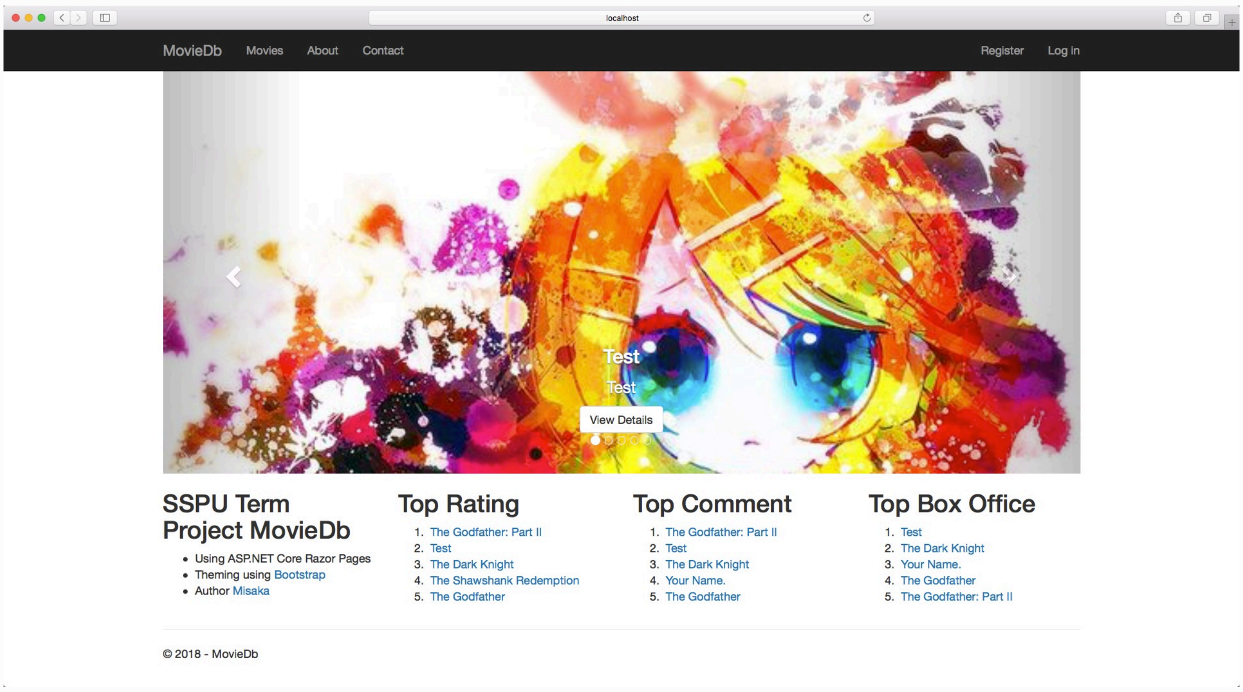
电影评分系统

2. 主要功能

由管理员增删改电影，用户对电影进行查询、评分和评论，可以显示电影的平均得分及各种信息。

二、系统（或网站）结构设计

1. 主页



SSPU Term Project MovieDb

- Using ASP.NET Core Razor Pages
- Theming using Bootstrap
- Author Misaka

Top Rating

1. The Godfather: Part II
2. Test
3. The Dark Knight
4. The Shawshank Redemption
5. The Godfather

Top Comment

1. The Godfather: Part II
2. Test
3. The Dark Knight
4. Your Name.
5. The Godfather

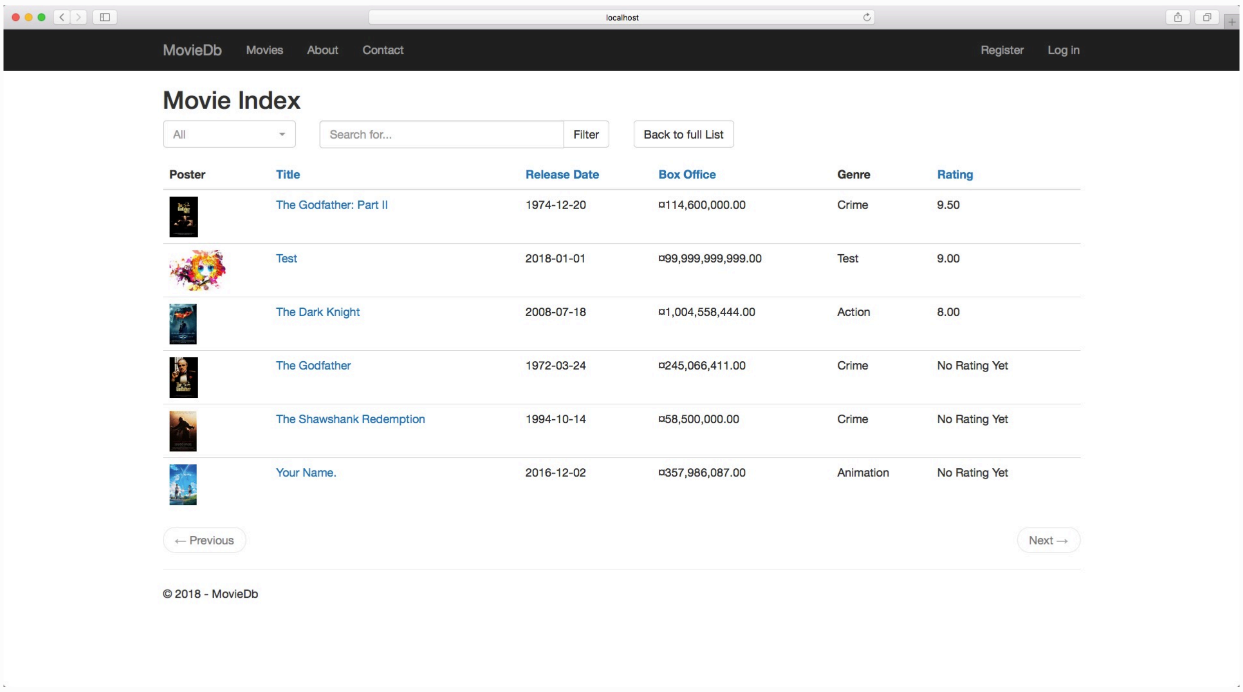
Top Box Office

1. Test
2. The Dark Knight
3. Your Name.
4. The Godfather
5. The Godfather: Part II

© 2018 - MovieDb

浏览最近发布的5部电影，滚动播放海报。
点击详情或链接可以跳转到电影的
浏览最新发布、评分最高、最多评论、最高票房 各5个；

2. 电影列表



Movie Index

All | Search for... | Filter | Back to full List

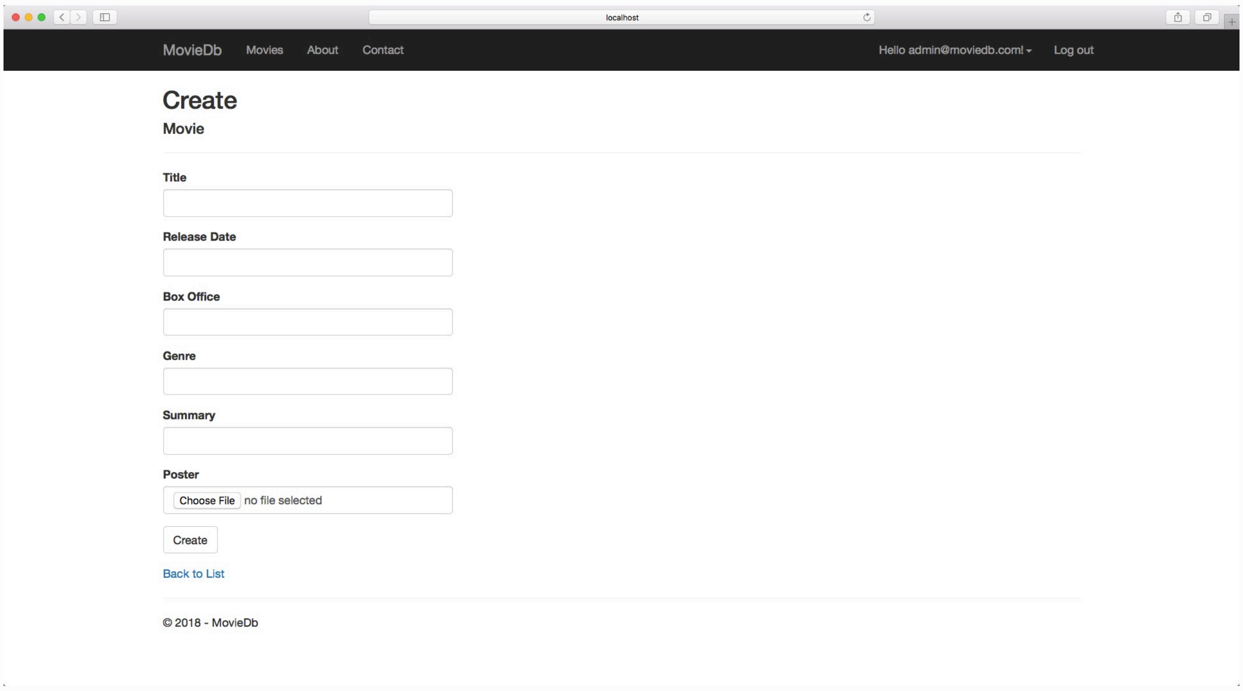
Poster	Title	Release Date	Box Office	Genre	Rating
	The Godfather: Part II	1974-12-20	¤114,600,000.00	Crime	9.50
	Test	2018-01-01	¤99,999,999,999.00	Test	9.00
	The Dark Knight	2008-07-18	¤1,004,558,444.00	Action	8.00
	The Godfather	1972-03-24	¤245,066,411.00	Crime	No Rating Yet
	The Shawshank Redemption	1994-10-14	¤58,500,000.00	Crime	No Rating Yet
	Your Name.	2016-12-02	¤357,986,087.00	Animation	No Rating Yet

← Previous   Next →

© 2018 - MovieDb

按标题和分类进行筛选、可以按标题，发布日期，票房，评分进行排序。

2.1 增加（管理员）



Create

Movie

Title

Release Date

Box Office

Genre

Summary

Poster

Choose File no file selected

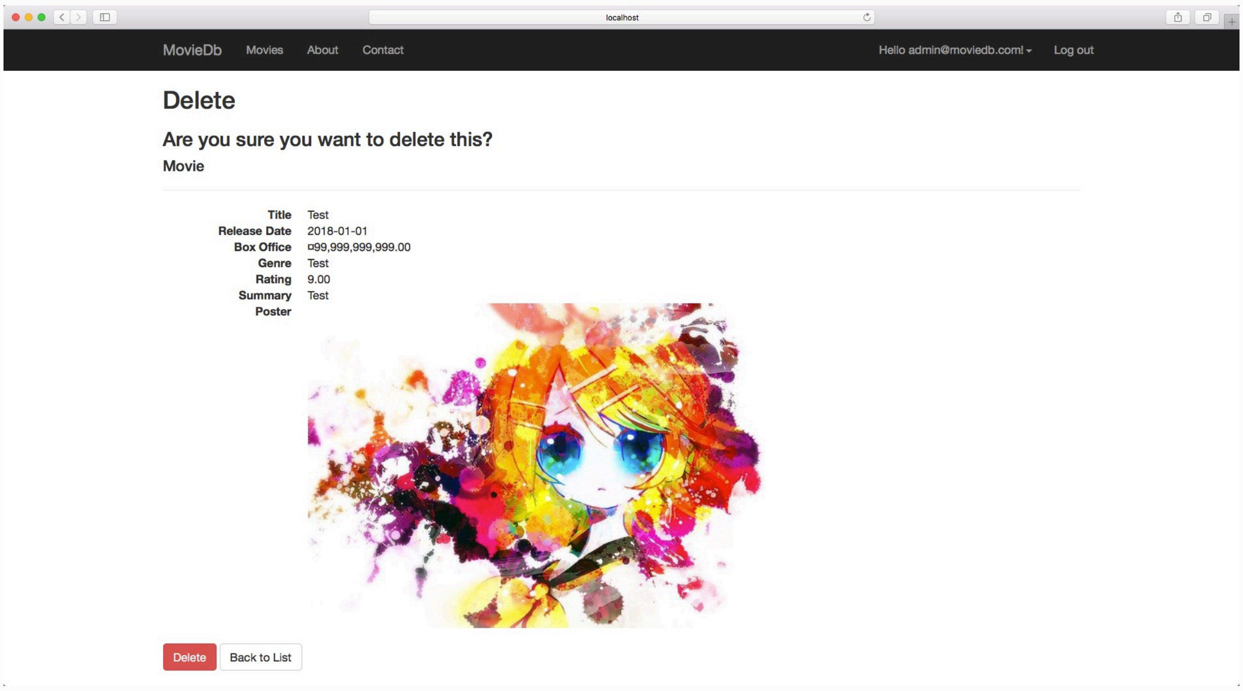
Create

Back to List

© 2018 - MovieDb

可以选择本地的图片，对各个字段可以验证

2.2 删除（管理员）


MovieDb Movies About Contact Hello admin@moviedb.coml Log out

Delete

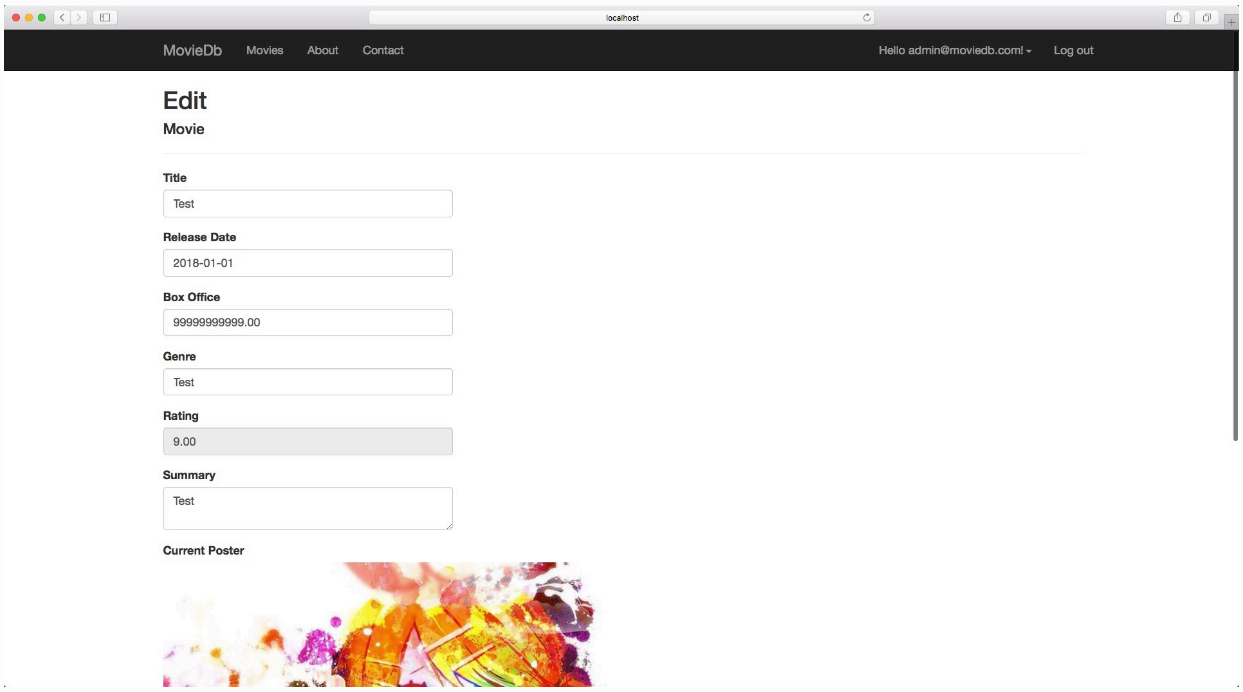
Are you sure you want to delete this?

Movie

Title	Test
Release Date	2018-01-01
Box Office	¤99,999,999,999.00
Genre	Test
Rating	9.00
Summary	Test
Poster	

先显示详细页面，确认后点击删除再删除

2.3 修改（管理员）

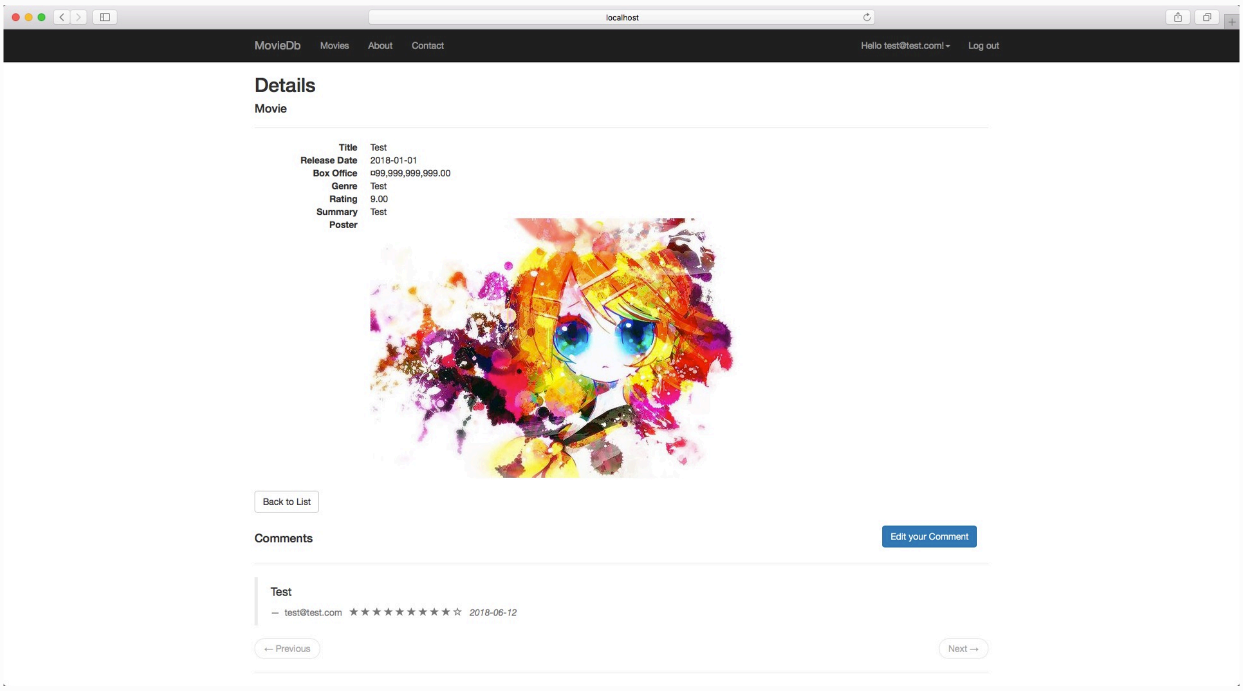

MovieDb Movies About Contact Hello admin@moviedb.coml Log out

Edit

Movie

Title	Test
Release Date	2018-01-01
Box Office	99999999999.00
Genre	Test
Rating	9.00
Summary	Test
Current Poster	

2.4 电影详情



Details

Movie

Title	Test
Release Date	2018-01-01
Box Office	¤99,999,999,999.00
Genre	Test
Rating	9.00
Summary	Test
Poster	

Comments

Test

— test@test.com ★★★★★★★★★☆ *2018-06-12*

下方显示评论和评分

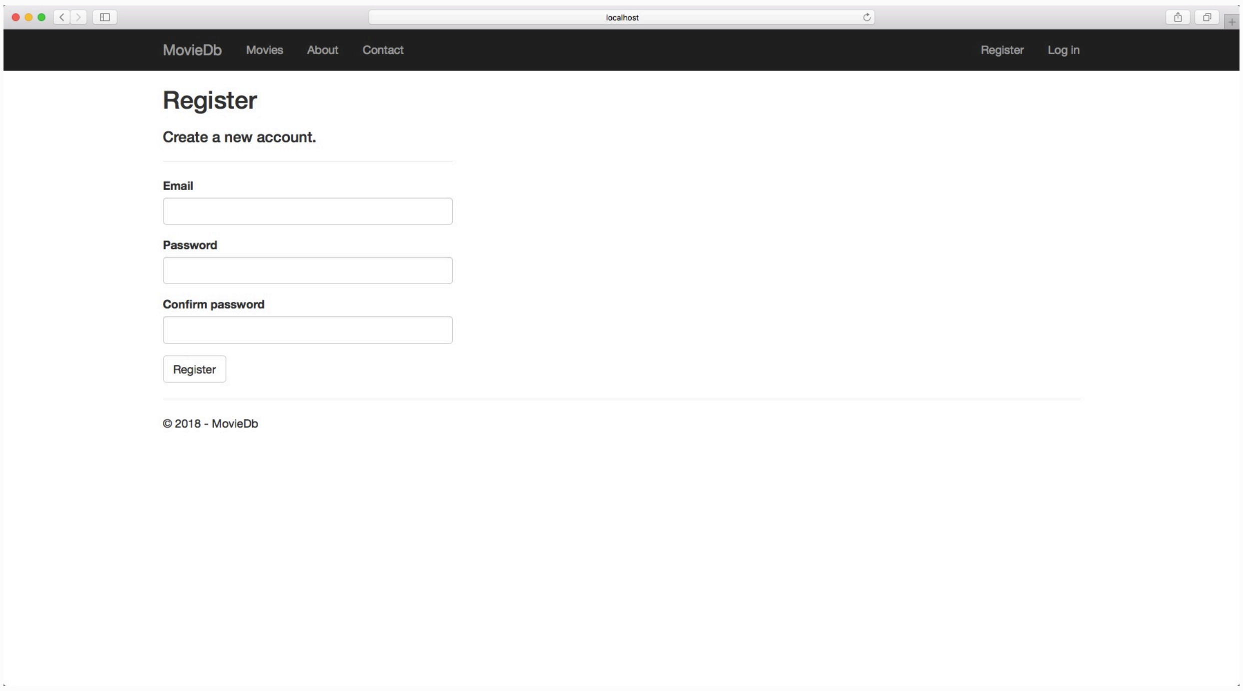
3. 关于

按照日期对电影分组

4. 联系

作者相关

5.注册



Register

Create a new account.

Email

Password

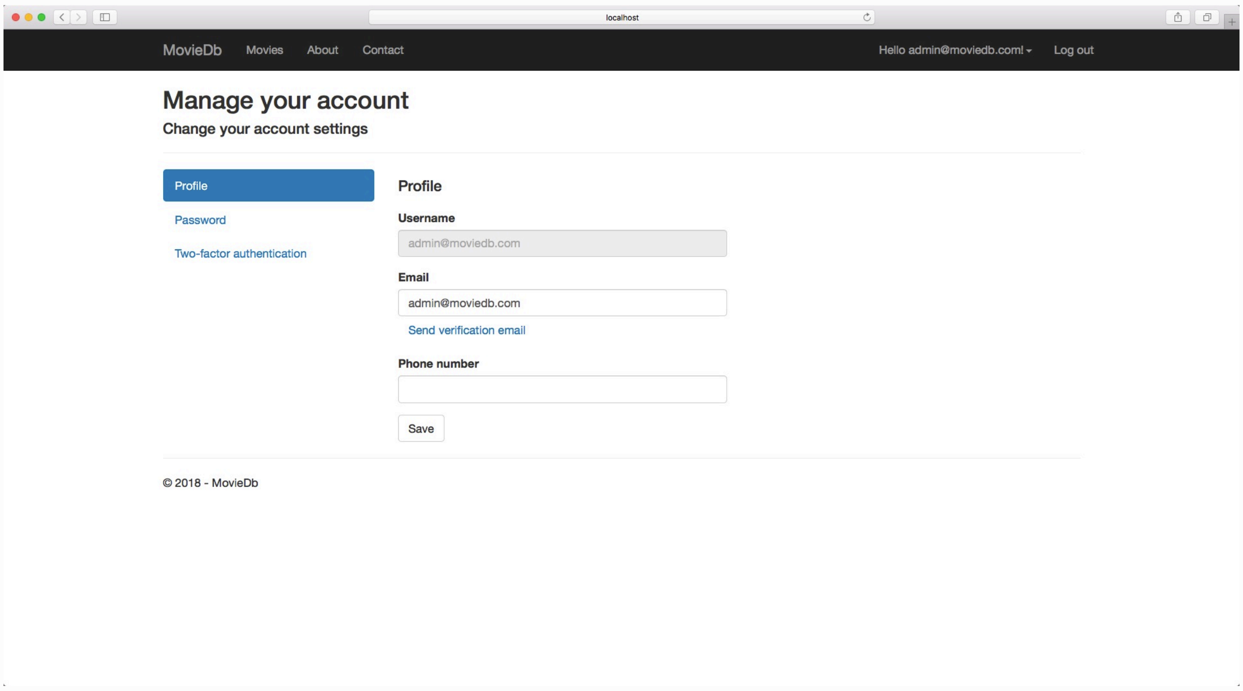
Confirm password

Register

© 2018 - MovieDb

对密码有强度验证

6. 用户



Manage your account

Change your account settings

Profile

Password

Two-factor authentication

Profile

Username

admin@moviedb.com

Email

admin@moviedb.com

Send verification email

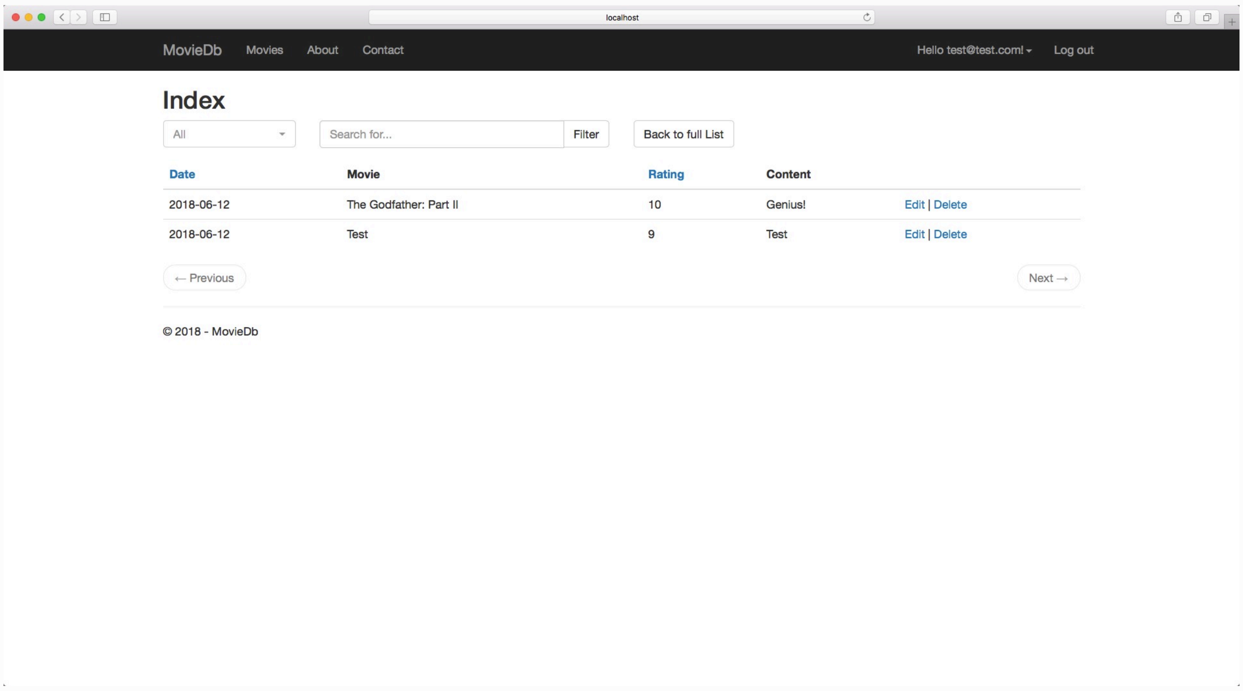
Phone number

Save

© 2018 - MovieDb

可以更改密码等

7. 评论列表


MovieDb Movies About Contact Hello test@test.com! Log out

Index

All | Search for... | Filter | Back to full List

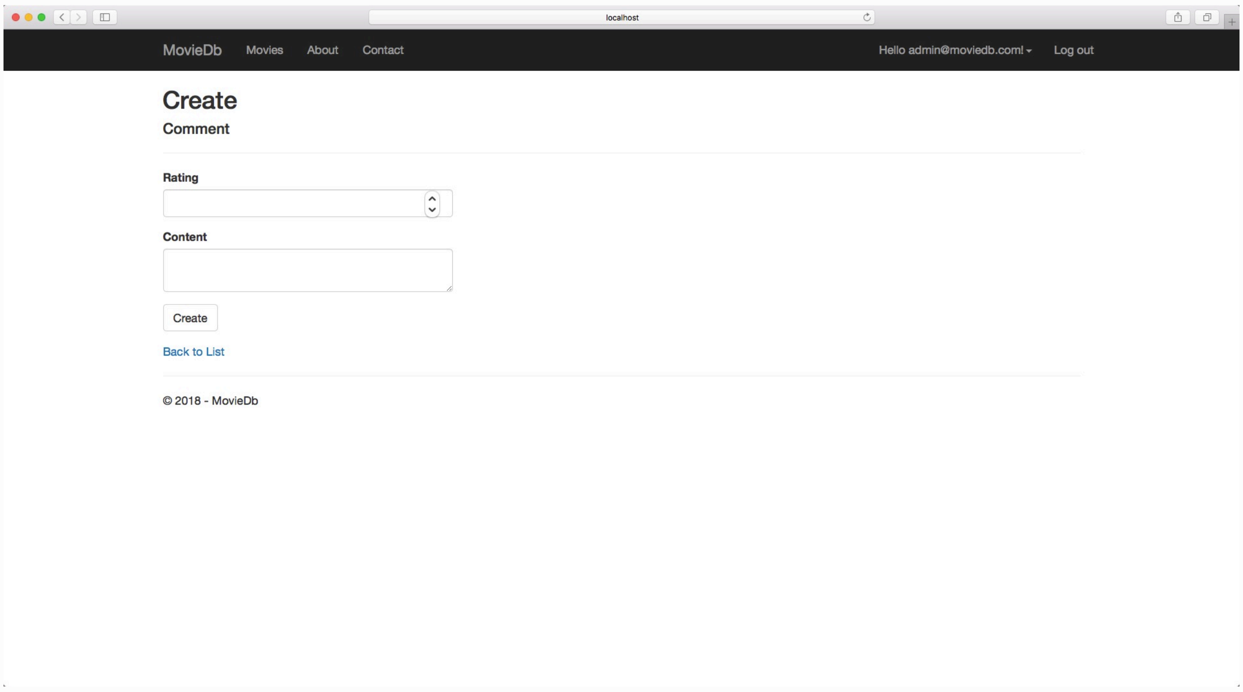
Date	Movie	Rating	Content	
2018-06-12	The Godfather: Part II	10	Genius!	Edit | Delete
2018-06-12	Test	9	Test	Edit | Delete

← Previous   Next →

© 2018 - MovieDb

用户可以编辑和删除自己的评论

7.1 增加


MovieDb Movies About Contact Hello admin@moviedb.com! Log out

Create

Comment

Rating

Content

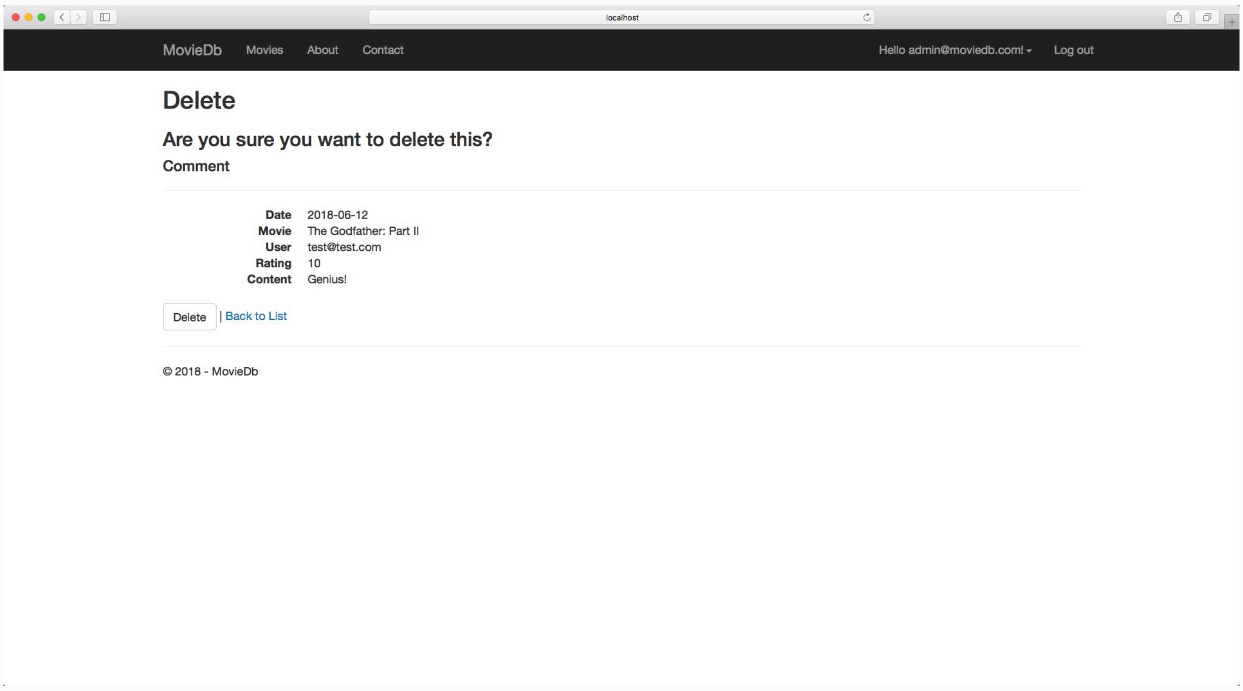
Create

Back to List

© 2018 - MovieDb

从电影详情页面进入

7.2 删除



Delete

Are you sure you want to delete this?

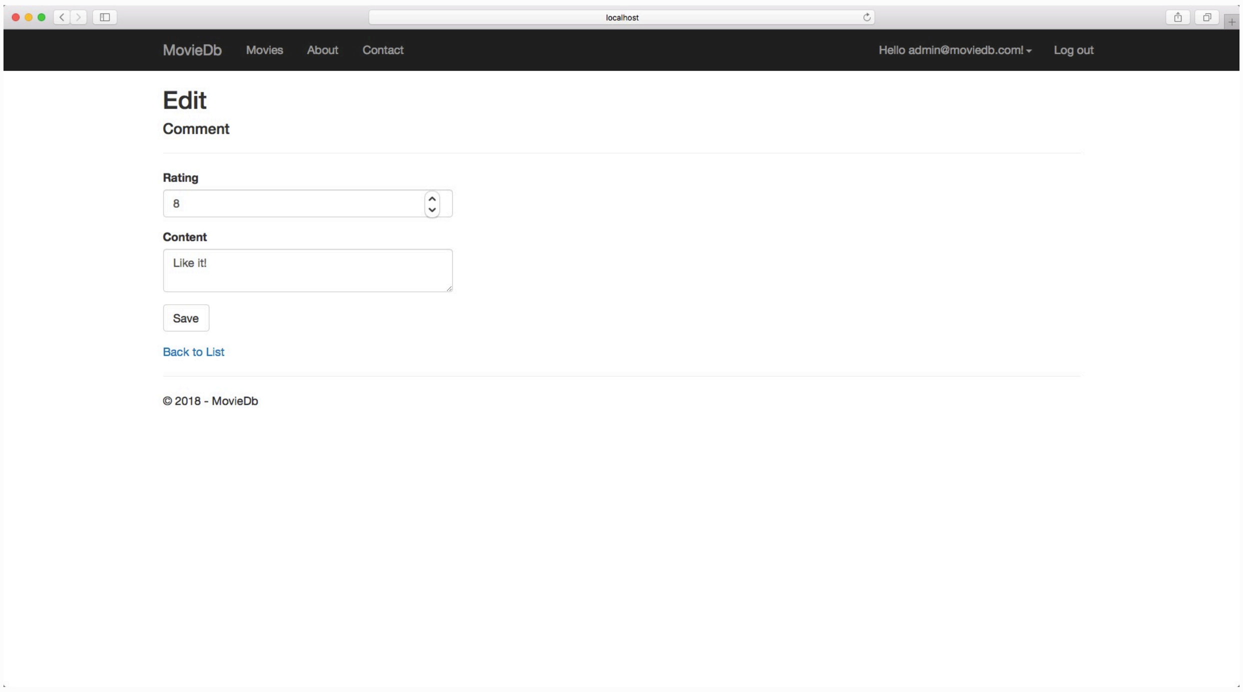
Comment

Date	2018-06-12
Movie	The Godfather: Part II
User	test@test.com
Rating	10
Content	Genius!

Delete | Back to List

© 2018 - MovieDb

7.3 更改



Edit

Comment

Rating: 8

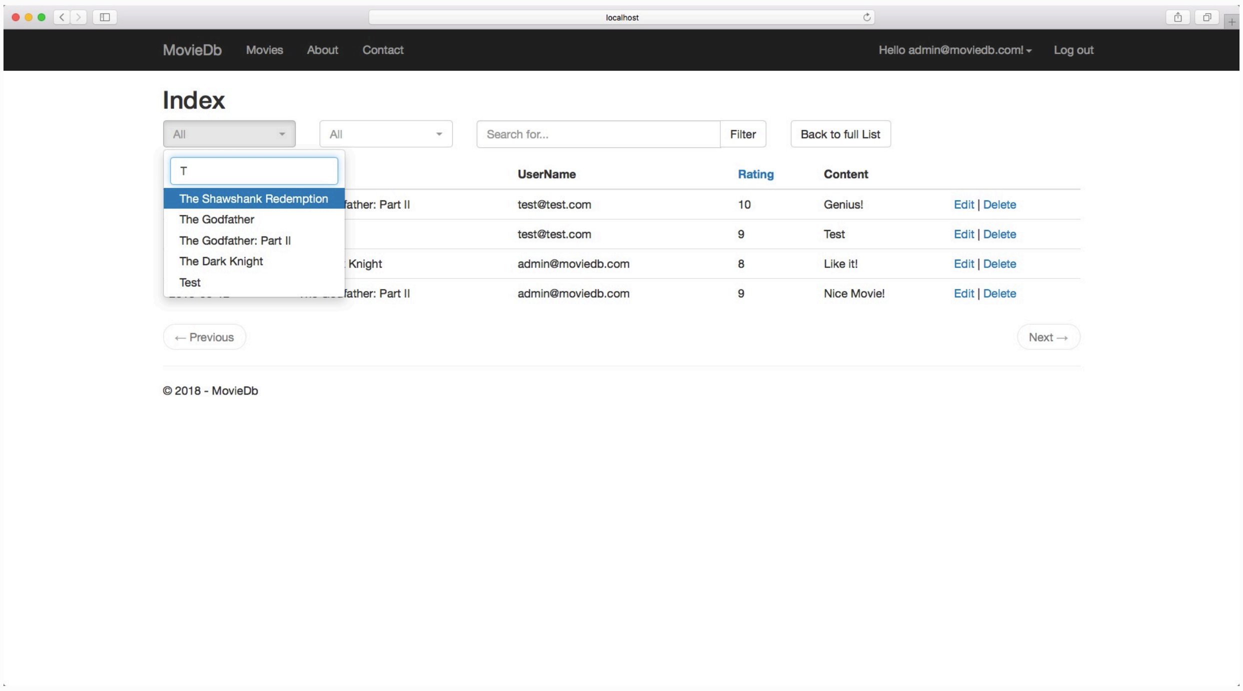
Content: Like it!

Save

Back to List

© 2018 - MovieDb

7.4 查询（管理员）



Index

All | All | Search for... | Filter | Back to full List

T

- The Shawshank Redemption
- The Godfather
- The Godfather: Part II
- The Dark Knight
- Test

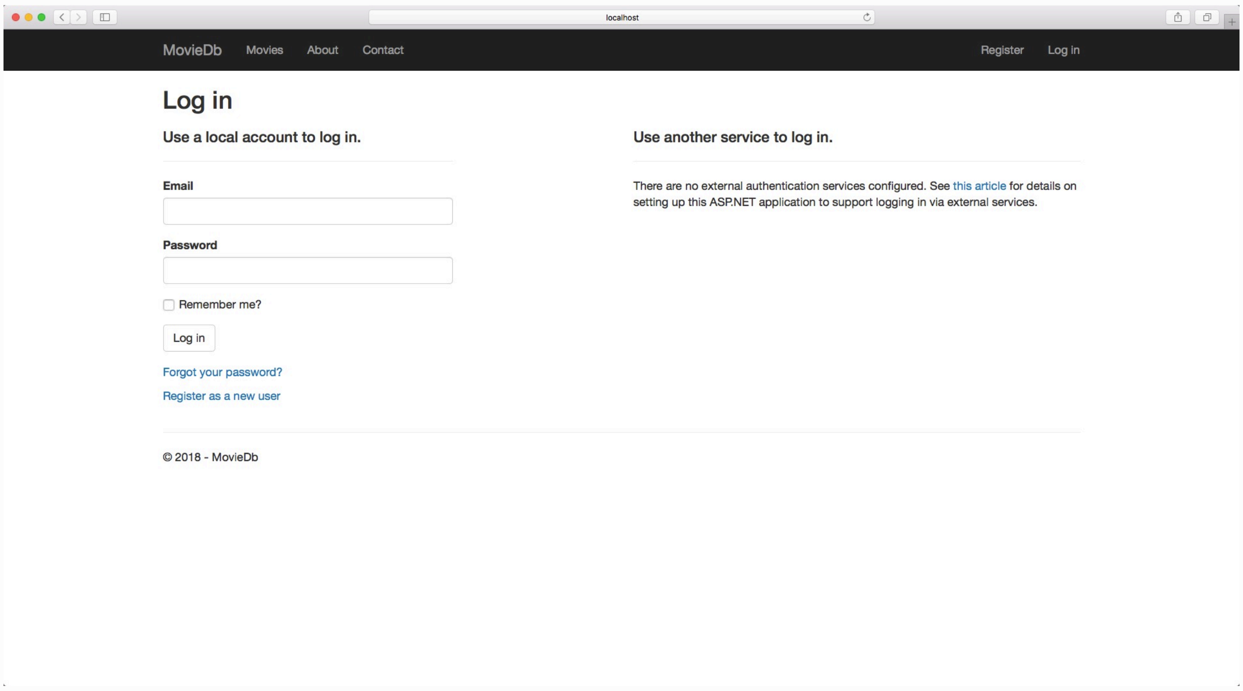
	UserName	Rating	Content	
father: Part II	test@test.com	10	Genius!	Edit | Delete
	test@test.com	9	Test	Edit | Delete
Knight	admin@moviedb.com	8	Like it!	Edit | Delete
father: Part II	admin@moviedb.com	9	Nice Movie!	Edit | Delete

← Previous　　Next →

© 2018 - MovieDb

可以根据电影和用户及其评论内容进行筛选

8. 登陆



Log in

Use a local account to log in.

Email

Password

Remember me?

Log in

Forgot your password?

Register as a new user

Use another service to log in.

There are no external authentication services configured. See this article for details on setting up this ASP.NET application to support logging in via external services.

© 2018 - MovieDb

对密码和用户名进行验证

9. 其他

以上所有界面都验证用户的身份，对于一切没有授权的行为都会跳转到登陆页面。
采用相应式布局，方便手机及平板用户。

三、代码说明

Data/Movie.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.ModelBinding;

namespace MovieDb.Data
{
    public class Movie
    {
        public int ID { get; set; }
        [StringLength(60, MinimumLength = 3)]
        [Required]
        public string Title { get; set; }

        [Display(Name = "Release Date")]
        [DataType(DataType.Date)]
        [DisplayFormat(DataFormatString = "{0:yyyy-MM-dd}", ApplyFormatInEditMode =
true)]
        public DateTime ReleaseDate { get; set; }

        [Display(Name = "Box Office")]
        [DataType(DataType.Currency)]
        public decimal BoxOffice { get; set; }

        [Required]
        [StringLength(30)]
        public string Genre { get; set; }

        [StringLength(50000)]
        [DataType(DataType.MultilineText)]
        public string Summary { get; set; }

        public byte[] Poster { get; set; }

        [NotMapped]
        public decimal Rating
        {
            get
            {
                if (Comments!=null && Comments.Count != 0)
                    return (decimal)Comments.Sum(c => c.Rating) / Comments.Count;
                else
                return -1;
            }
        }

        [NotMapped]
        public int CommentsCount
        {
            get
            {
                if (Comments != null)
                    return Comments.Count;
                else return 0;
            }
        }

        public ICollection<Comment> Comments { get; set; }
    }
}
```

模型类 各个字段都进行约束，在输入和编辑时可以进行验证，海报字段使用byte[]数组存储，可以存入数据库中，DataType字段设定后可以调用浏览器都H5特性，如日历等。 评分字段为只读，是计算列，通过多表链接查询得出。

_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewData["Title"] - MovieDb</title>

    <environment include="Development">
        <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.css" />
        <link rel="stylesheet" href="~/css/site.css" />
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
select/1.12.4/css/bootstrap-select.min.css">
        <link rel="stylesheet" href="https://cdn.bootcss.com/object-fit/0.4.3/polyfill.object-
fit.min.css">
    </environment>
    <environment exclude="Development">
        <link rel="stylesheet"
href="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/css/bootstrap.min.css"
              asp-fallback-href="~/lib/bootstrap/dist/css/bootstrap.min.css"
              asp-fallback-test-class="sr-only" asp-fallback-test-property="position" asp-
fallback-test-value="absolute" />
```

```
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
select/1.12.4/css/bootstrap-select.min.css">
        <link rel="stylesheet" href="https://cdn.bootcss.com/object-fit/0.4.3/polyfill.object-
fit.min.css">
        <link rel="stylesheet" href="~/css/site.min.css" asp-append-version="true" />
    </environment>
</head>
<body>
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target=".navbar-collapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a asp-page="/Index" class="navbar-brand">MovieDb</a>
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">
                    <li><a asp-page="/Movies/Index">Movies</a></li>
                    <li><a asp-page="/About">About</a></li>
                    <li><a asp-page="/Contact">Contact</a></li>
                </ul>
                @await Html.PartialAsync("_LoginPartial")
            </div>
        </div>
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; 2018 - MovieDb</p>
        </footer>
    </div>

    <environment include="Development">
        <script src="~/lib/jquery/dist/jquery.js"></script>
        <script src="~/lib/bootstrap/dist/js/bootstrap.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
select/1.12.4/js/bootstrap-select.min.js"></script>
        <script src="https://cdn.bootcss.com/object-fit/0.4.3/polyfill.object-fit.min.js">
</script>
        <script src="~/js/site.js" asp-append-version="true"></script>
    </environment>
    <environment exclude="Development">
        <script src="https://ajax.aspnetcdn.com/ajax/jquery/jquery-2.2.0.min.js"
                asp-fallback-src="~/lib/jquery/dist/jquery.min.js"
                asp-fallback-test="window.jQuery"
                crossorigin="anonymous"
            integrity="sha384-
K+ctZQ+LL8q6tP7I94W+qzQsfRV2a+AfHIi9k8z8l9ggpc8X+Ytst4yBo/hH+8Fk"></script>
        <script src="https://ajax.aspnetcdn.com/ajax/bootstrap/3.3.7/bootstrap.min.js"
            asp-fallback-src="~/lib/bootstrap/dist/js/bootstrap.min.js"
            asp-fallback-test="window.jQuery && window.jQuery.fn &&
window.jQuery.fn.modal"
            crossorigin="anonymous"
            integrity="sha384-
Tc5IQib027qvyjSMfHjOMaLkfuWVxZxUPnCJA7l2mCWNIpG9mGCD8wGNIcPD7Txa">
</script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
select/1.12.4/js/bootstrap-select.min.js"></script>
        <script src="https://cdn.bootcss.com/object-fit/0.4.3/polyfill.object-fit.min.js">
</script>
        <script src="~/js/site.min.js" asp-append-version="true"></script>
    </environment>

    @RenderSection("Scripts", required: false)
</body>
</html>
```

类似于ASP.NET中的模板页面，确定了页面头部和尾部，同时包含了bootstrap的css和js引用

Comments/Index.cshtml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.RazorPages;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using MovieDb.Authorization;
using MovieDb.Data;

namespace MovieDb.Pages.Comments
{
    public class IndexModel : PageModel
    {
        private readonly MovieDb.Data.ApplicationDbContext _context;

        private readonly UserManager<ApplicationUser> _userManager;

        public IndexModel(MovieDb.Data.ApplicationDbContext context,
                    UserManager<ApplicationUser> userManager)
        {
            _context = context;
        _userManager = userManager;
    }

    public PaginatedList<Comment> Comment { get; set; }
    public SelectList Movies { get; set; }
    public SelectList Users { get; set; }

    public string DateSort { get; set; }
    public string RatingSort { get; set; }
    public string CurrentFilter { get; set; }
    public string CurrentSort { get; set; }
    public int? CurrentMovieID { get; set; }
    public string CurrentUserID { get; set; }

    public async Task OnGetAsync(string sortOrder,
                                 string currentFilter,
                                 string searchString,
                                 int? currentMovieID,
                                 int? movieID,
                                 string currentUserID,
                                 string userID,
                                 int? pageIndex)
    {
        CurrentSort = sortOrder;
        DateSort = sortOrder == "Comment Date" ? "date_desc" : "Comment Date";
        RatingSort = sortOrder == "Rating" ? "rating_desc" : "Rating";

        if (searchString != null)
        {
            pageIndex = 1;
        }
        else
        {
            searchString = currentFilter;
        }
        if (movieID != null)
        {
            pageIndex = 1;
        }
        else
        {
            movieID = currentMovieID;
        }
        if (userID != null)
        {
            pageIndex = 1;
        }
        else
        {
            userID = currentUserID;
        }
        CurrentFilter = searchString;
```

```
            CurrentUserID = userID;
            CurrentMovieID = movieID;

            IQueryable<Movie> movies = from m in _context.Movie
                                       select m;

            IQueryable<ApplicationUser> users = from u in _context.Users
                                                select u;

            IQueryable<Comment> comments = from c in
_context.Comment.Include("Movie")
                                              .Include("User")
                                            select c;

            if (!(User.IsInRole(Constants.MovieAdministratorsRole) ||
User.IsInRole(Constants.MovieManagersRole)))
            {
                comments = comments.Where(c =>
c.UserID.Equals(_userManager.GetUserId(User)));
            }

            if (!String.IsNullOrEmpty(searchString))
            {
                comments = comments.Where(c => c.Content.Contains(searchString));
            }

            if (movieID != null && movieID != 0)
            {
                comments = comments.Where(c => c.MovieID.Equals(movieID));
            }

            if (!String.IsNullOrEmpty(userID))
            {
                comments = comments.Where(c => c.UserID.Equals(userID));
            }

            switch (sortOrder)
            {
                case "Comment Date":
                    comments = comments.OrderBy(c => c.Date);
                    break;
                case "date_desc":
                    comments = comments.OrderByDescending(c => c.Date);
                    break;
                case "Rating":
                    comments = comments.OrderBy(c => c.Rating);
                    break;
                case "rating_desc":
                    comments = comments.OrderByDescending(c => c.Rating);
                    break;
                default:
                    comments = comments.OrderByDescending(c => c.Date);
                    break;
                }
                Movies = new SelectList(movies, "ID", "Title");

                Users = new SelectList(users, "Id", "UserName");

                int pageSize = 6;
                Comment = await PaginatedList<Comment>.CreateAsync(comments,
pageIndex ?? 1, pageSize);
            }
        }
}
```

实现评论类表页面的筛选、排序和分页，`lambda` 表达式和 `linq` 用来指定从数据库中查询筛选的内容，async是系统的异步，`IQueryable` 是在最终生成List时才从数据库中取数据到内存中，而 `IEnumerable` 在查询时就取数据到内存中了，所以在对数据库中没有存的列进行排序时（如类似的电影列表），需要使用 `IEnumerable` 。

Movies/Index.cshtml

```
@page
@model MovieDb.Pages.Movies.IndexModel

@{
    ViewData("Title") = "Movie Index";
}

<h2>Movie Index</h2>

@if (User.IsInRole(Constants.MovieAdministratorsRole) | |
User.IsInRole(Constants.MovieManagersRole))
{
    <p>
        <a asp-page="Create">Create New</a>
    </p>
}

<form asp-page="./Index" method="get">
    <div class="form-actions">
        <div class="form-group row">
            <div class="col-lg-2 col-md-2 col-sm-2 col-xs-4">
                <select class="selectpicker form-control" asp-for="CurrentMovieGenre" asp-
items="Model.Genres">
                    <option value="">All</option>
                    <option data-divider="true"></option>
                </select>
            </div>
            <div class="col-lg-4 col-md-4 col-sm-4 col-xs-6">
                <div class="input-group">
```

```
                    <input class="form-control" placeholder="Search for..." type="text"
name="SearchString" value="@Model.CurrentFilter" />
                    <span class="input-group-btn">
                        <button class="btn btn-default" type="submit">Filter</button>
                    </span>
                </div>
            </div>
            <div class="col-lg-2 col-md-2 col-sm-2 col-xs-4">
                <a class="btn btn-default" asp-page="./Index">Back to full List</a>
            </div>
        </div>
    </div>
</form>

<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Movie[0].Poster)
            </th>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.TitleSort"
                   asp-route-currentFilter="@Model.CurrentFilter"
                   asp-route-currentMovieGenre="@Model.CurrentMovieGenre">
                    @Html.DisplayNameFor(model => model.Movie[0].Title)
                </a>
            </th>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.DateSort"
                   asp-route-currentFilter="@Model.CurrentFilter"
                   asp-route-currentMovieGenre="@Model.CurrentMovieGenre">
                    @Html.DisplayNameFor(model => model.Movie[0].ReleaseDate)
                </a>
            </th>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.BoxOfficeSort"
                   asp-route-currentFilter="@Model.CurrentFilter"
                   asp-route-currentMovieGenre="@Model.CurrentMovieGenre">
                    @Html.DisplayNameFor(model => model.Movie[0].BoxOffice)
                </a>
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Movie[0].Genre)
            </th>
            <th>
                <a asp-page="./Index" asp-route-sortOrder="@Model.RatingSort"
                   asp-route-currentFilter="@Model.CurrentFilter"
                   asp-route-currentMovieGenre="@Model.CurrentMovieGenre">
                    @Html.DisplayNameFor(model => model.Movie[0].Rating)
                </a>
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model.Movie)
        {
            <tr>
                <td>
                    @if (item.Poster != null)
                    {
                        var base64 = Convert.ToBase64String(item.Poster);
                        var imgsrc = string.Format("data:image/jpg;base64,{0}", base64);
                        <img src="@imgsrc" height="50" />
                    }
                </td>
                <td>
                    <a asp-page="./Details" asp-route-id="@item.ID">
                        @Html.DisplayFor(modelItem => item.Title)
                    </a>
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.ReleaseDate)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.BoxOffice)
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.Genre)
                </td>
                <td>
                    @if (item.Rating == -1)
                    {
                        <text>No Rating Yet</text>
                    }
                    else
                    {
                        @Html.DisplayFor(modelItem => item.Rating);
                    }
                </td>
                <td>
                    @if ((await AuthorizationService.AuthorizeAsync(User, item,
EntityOperations.Update)).Succeeded)
                    {
                        <a asp-page="./Edit" class="btn btn-warning btn-sm" asp-route-
id="@item.ID">Edit</a>
                    }

                    @if ((await AuthorizationService.AuthorizeAsync(User, item,
EntityOperations.Delete)).Succeeded)
                    {
                        <a asp-page="./Delete" class="btn btn-danger btn-sm" asp-route-
id="@item.ID">Delete</a>
                    }
                </td>
            </tr>
        }
    </tbody>
</table>
@{
    var prevDisabled = !Model.Movie.HasPreviousPage ? "disabled" : "";
    var nextDisabled = !Model.Movie.HasNextPage ? "disabled" : "";
}
<nav aria-label="...">
    <ul class="pager">
        <li class="previous">
            <a asp-page="./Index"
               asp-route-sortOrder="@Model.CurrentSort"
               asp-route-pageIndex="@(Model.Movie.PageIndex - 1)"
               asp-route-currentFilter="@Model.CurrentFilter"
               asp-route-currentMovieGenre="@Model.CurrentMovieGenre"
               class="btn btn-default @prevDisabled">
                <span aria-hidden="true">&larr;</span>
                Previous
            </a>
        </li>
        <li class="next">
            <a asp-page="./Index"
               asp-route-sortOrder="@Model.CurrentSort"
               asp-route-pageIndex="@(Model.Movie.PageIndex + 1)"
               asp-route-currentFilter="@Model.CurrentFilter"
               asp-route-currentMovieGenre="@Model.CurrentMovieGenre"
               class="btn btn-default @nextDisabled">
                Next
                <span aria-hidden="true">&rarr;</span>
            </a>
        </li>
    </ul>
</nav>
```

图片使用base64编码从数据库中读取，最下方为分页的导航按钮，在每次点击刷新需要跳转页面时需要将当前的筛选值保存下来，并在Get时将其加载出来。

`Authorization` 文件夹中声明了两个角色和对应的权限，用于页面的验证身份。

定义了模型类后使用 `EF Core` 进行数据库的生成。

四、设计小结（体会）

为了紧跟时代的步伐，在设计大作业时尝试了之前从未踏足的ASP.NET Core ，并得到了老师的支持，在此深表感谢。

目前国内.Net Core的资源并不多，所以一切都是从头做起，但也感受到新框架的便利性（开源、跨平台、Razor页面中调用C#、依赖注入等）相信在不久的将来这会成为主流。

在设计和编码的过程中，经常能遇到一些细节的问题，这就关系到基本的C#编码基础和对网页三剑客(Html5、CSS、JavaScript)的理解了，如响应式布局，各个不同用户角色的不同元素显示，lambda表达式，文件的处理和C#中不同List的用法，前端框架中提供的一些布局等等。

在设计模型时，由于和数据库是同步的，所以要考虑外键以及实体之间的关系，在处理图片填充的时候，需要有对盒式布局的理解，在数据库查询的时候，需要有对linq和EF Core的理解。

可以说虽然框架换了但是网站的核心还是C#和三剑客，即一个好用的后端语言和前端框架，可谓掌握了这些基础的内容之后，也就方便了许多，可见基础的重要性。