

.net 架构第四次大作业

目录

第一题:	2
二、项目文件简要概述.....	3
三、项目功能具体实现.....	4
四、程序运行截图.....	7
五、遇到的问题与解决.....	8
六、代码附录.....	9

第一题：

一、要求：

假设购物车 Cart 表字段如下：

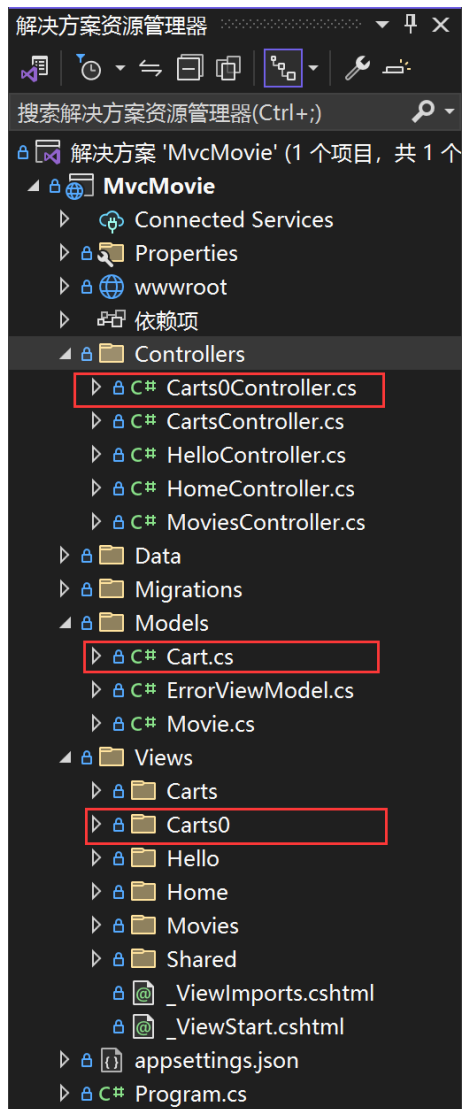
cart_id(购物车 id)、product_id(商品 id)、product_num(商品数量)、product_price(商品价格)、user_id(用户 id)、createtime(创建时间)等字段。

参考所学 Movie 示例，完成：

- (1) 根据 Cart 表字段设计购物车 Cart 模型类。
- (2) 使用“新搭建基架的项目”功能创建控制器，完成基本 CRUD 功能。
- (3) 在 Index 首页中，添加查询：根据 product_id 或 product_price 区间查询。

二、项目文件简要概述

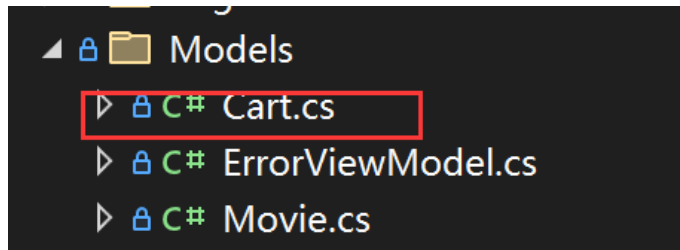
在原有的 Movie 项目中加入 cart 部分



红色部分标注 Cart 的 Model, View 和 Controller 部分

三、项目功能具体实现

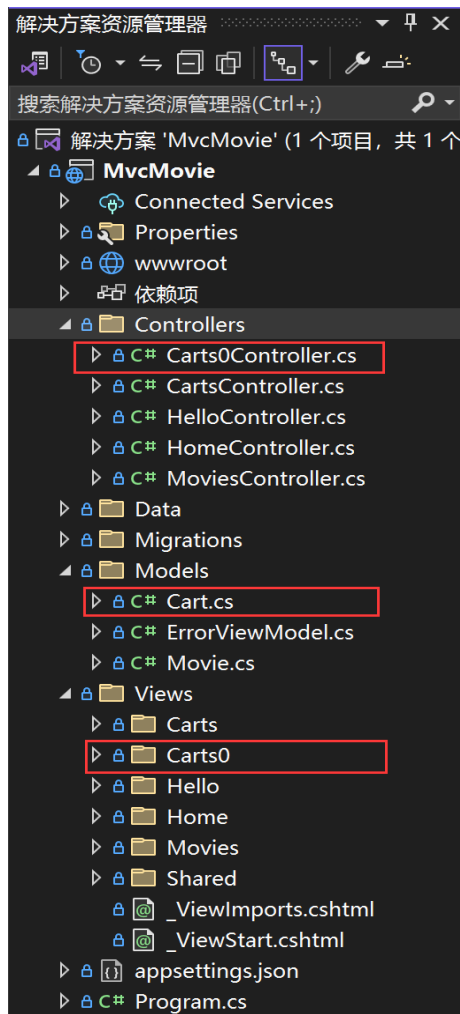
1、新建 Cart.cs 模型



2、在模型中添加属性，并且给属性添加注解、

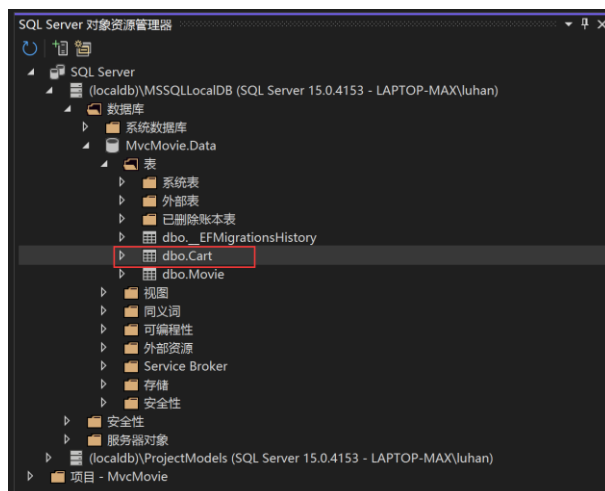
```
1 using System.ComponentModel.DataAnnotations;
2
3 namespace MvcMovie.Models
4 {
5     35 个引用
6     public class Cart
7     {
8         22 个引用
9         public int Id { get; set; } //Cart_id
10
11         [Required(ErrorMessage = "购物车id必填")]
12         [StringLength(20, MinimumLength = 1, ErrorMessage = "1-20个字符")]
13         [Display(Name = "购物车id")]
14         12 个引用
15         public string? cart_id { get; set; } //购物车id
16
17         [Required(ErrorMessage = "商品id必填")]
18         [StringLength(20, MinimumLength = 1, ErrorMessage = "1-20个字符")]
19         [Display(Name = "商品id")]
20         25 个引用
21         public string? product_id { get; set; } //商品id
22
23         [Required(ErrorMessage = "商品数量必填")]
24         [Range(1, 100, ErrorMessage = "1-100之间")]
25         [Display(Name = "商品数量")]
26         [DataType(DataType.Currency)]
27         24 个引用
28         public int product_num { get; set; } //商品数量
29
30         [Required(ErrorMessage = "商品价格必填")]
31         [Range(1, 100, ErrorMessage = "1-100之间")]
32         [Display(Name = "商品价格")]
33         [DataType(DataType.Currency)]
34         26 个引用
35         public decimal product_price { get; set; } //商品价格
36
37         [Required(ErrorMessage = "用户id必填")]
38         [StringLength(20, MinimumLength = 1, ErrorMessage = "1-20个字符")]
39         [Display(Name = "用户id")]
40         24 个引用
41         public string? user_id { get; set; } //用户id
42
43         [Required(ErrorMessage = "创建时间必填")]
44         [Display(Name = "创建时间")]
45         [DataType(DataType.Date)]
46         24 个引用
47         public DateTime createtime { get; set; } //创建时间
48     }
49 }
```

3、使用 EF Core 框架为自动模型生成 CRUD 功能（Create, Read, Update 和 Delete）
自动生成的控制器和视图可以在下图中看到：

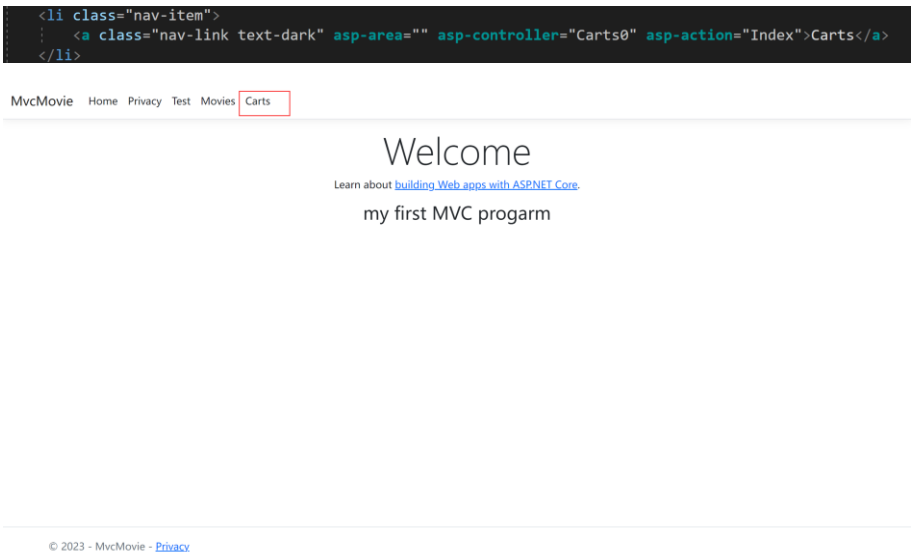


4、EF Core 迁移 – 创建数据库

依次使用① Add-Migration InitialCreate ② Update-Database，迁移成功且更新数据库后，可以在 SQL Server 对象资源管理器中看到建立好的数据库以及新建的表：



5、在导航栏中加入 Carts 选项方便访问：



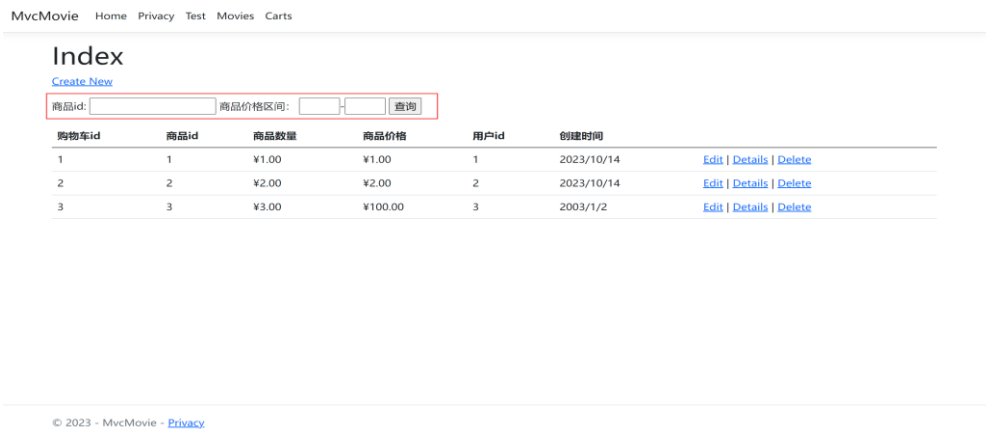
至此，已经可以实现基本的增删改查功能了，下面将加入根据 product_id 和 product_price 区间查询选项。

6、在控制器中加入控制逻辑，其中使用了异步编程模式：

```
3 个引用
public async Task<IActionResult> Index(string searchstring, double? low, double? high)
{
    if (String.IsNullOrEmpty(searchstring)) { searchstring = ""; }
    if (low == null) { low = 0.0; }
    if (high == null) { high = 999.0; }
    var carts = from m in _context.Cart
                where m.product_id.Contains(searchstring) && Convert.ToDecimal(low) <= m.product_price
                  && m.product_price <= Convert.ToDecimal(high)
                select m;
    return View(await carts.ToListAsync());
}
```

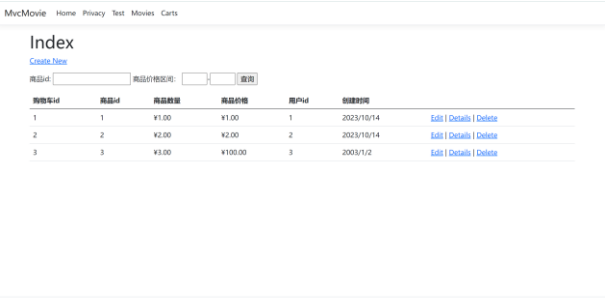
7、修改 Index 视图：

```
<form asp-controller="Carts0" asp-action="Index" method="get">
    <p>
        商品id:
        <input type="text" name="SearchString">
        商品价格区间:
        <input type="text" name="low" style="width:60px;">-<input type="text" name="high" style="width:60px;">
        <input type="submit" value="查询" />
    </p>
</form>
```

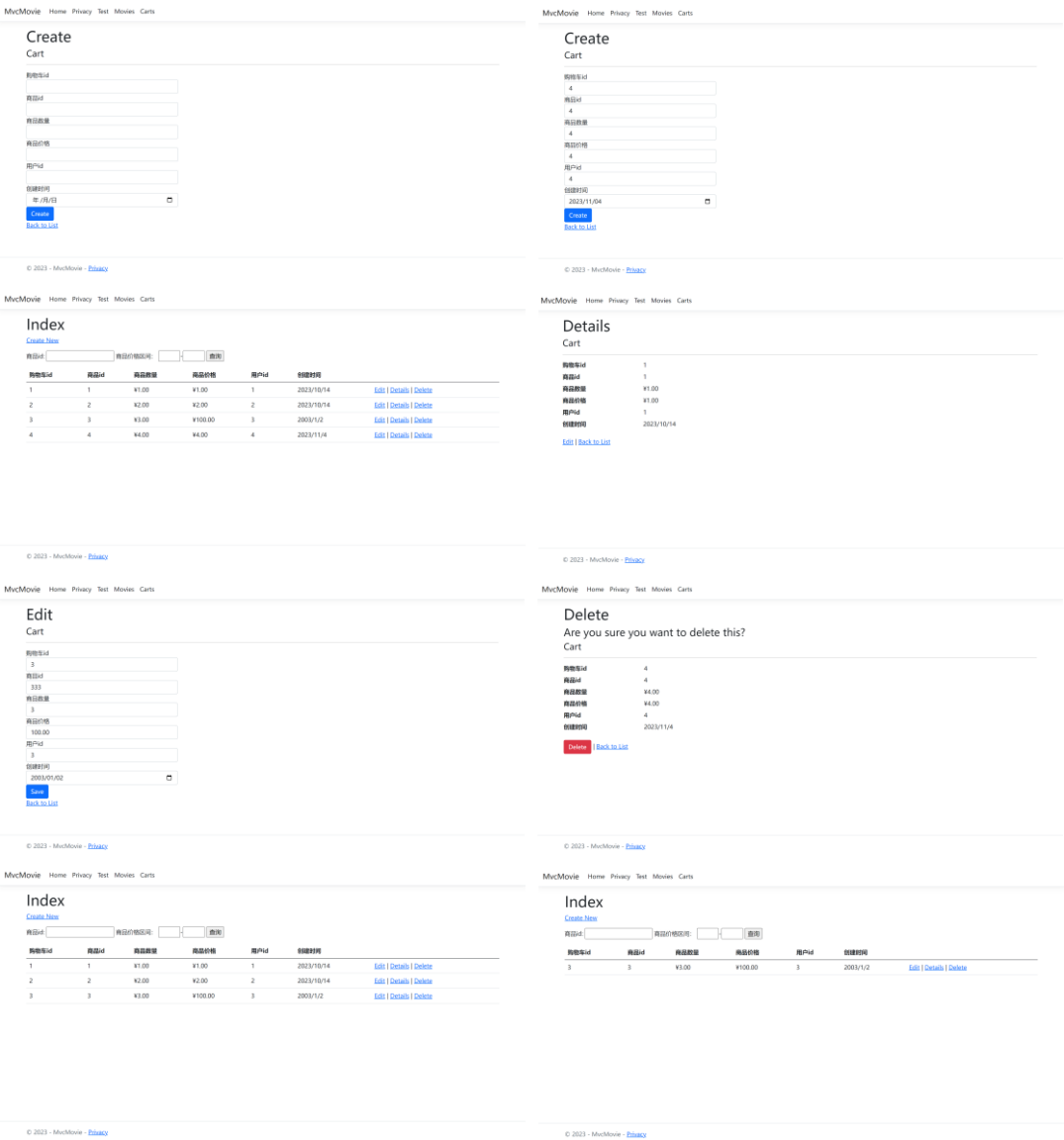


四、程序运行截图

基础界面：



增删改查：



具体可以登录 https://github.com/CLBDML/luhanbo_net_homework4 查看

五、遇到的问题与解决

已经解决的问题

在使用 Add-Migration InitialCreate 创建数据库迁移时候，因为在实现 movies 功能时候以及创建过一个数据库了，会出现报错

```
hasConversion .
```

```
The name 'InitialCreate' is used by an existing migration.
```

```
PM>
```

使用 Add-Migration InitialCreate0 或者其他名称避免重名即可顺利完成创建数据库迁移

六、代码附录

工程文件详见 https://github.com/CLBDML/luhanbo_net_homework4