**Платформа разработки Веб-приложений ASP.NET**

**Задание 1**

**Каталог фильмов**

* **ViewModel**: создать MovieViewModel с названием, жанром, годом, рейтингом.
* **Форма + валидация**: форма добавления фильма, поля [Required], [Range], [StringLength].
* **Сервис**: реализовать IMovieService с методами добавления и получения фильмов.
* **DI**: внедрить MovieService в MovieController.
* **TempData/ViewBag**: сообщение об успешном добавлении фильма через TempData.

**Код**

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

// Удаляем Razor Pages, оставляем только MVC

// builder.Services.AddRazorPages();

builder.Services.AddControllersWithViews();

// Register MovieService

builder.Services.AddSingleton<WebApplication1.Services.IMovieService, WebApplication1.Services.MovieService>();

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

// Простой редирект с корневого маршрута на Movie/Index

app.Use(async (context, next) =>

{

if (context.Request.Path.Value == "/")

{

context.Response.Redirect("/Movie");

return;

}

await next();

});

// Стандартный маршрут для контроллеров

app.MapControllerRoute(

name: "default",

pattern: "{controller=Movie}/{action=Index}/{id?}");

// Удаляем маппинг Razor Pages

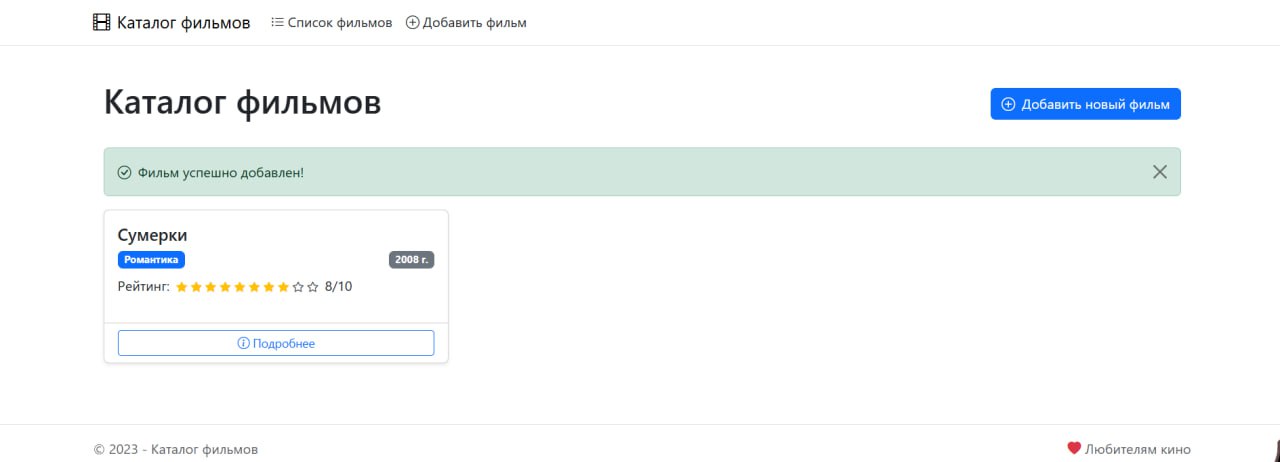
// app.MapRazorPages();

app.Run();

**Таблица 21 – Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **Сумерки 2008 Романтика 8** | **Сумерки 2008 Романтика 8** |

**Анализ результата**

****