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

Задание 1. Библиотека книг:

* **ViewModel: BookViewModel с названием, автором, ISBN, жанром.**
* **Форма + валидация: все поля обязательны, ISBN — [RegularExpression].**
* **Сервис: IBookService с методами получения списка, добавления.**
* **DI: внедрить сервис в BookController.**
* **TempData/ViewBag**: сообщение о новой книге через TempData.

Листинг программы:

namespace BookLibrary.Controllers

{

public class BooksController : Controller

{

private readonly IBookService \_bookService;

public BooksController(IBookService bookService)

{

\_bookService = bookService;

}

public IActionResult Index()

{

var books = \_bookService.GetAllBooks().ToList(); // Явное преобразование

return View(books);

}

public IActionResult Create()

{

return View();

}

public IActionResult Details(int id)

{

var book = \_bookService.GetAllBooks().FirstOrDefault(b => b.Id == id);

if (book == null)

{

return NotFound();

}

return View(book);

}

public IActionResult Edit(int id)

{

var book = \_bookService.GetAllBooks().FirstOrDefault(b => b.Id == id);

if (book == null)

{

return NotFound();

}

var model = new BookViewModel

{

Id = book.Id,

Title = book.Title,

Author = book.Author,

ISBN = book.ISBN,

Genre = book.Genre,

Year = book.Year

};

return View(model);

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Edit(BookViewModel model)

{

if (!ModelState.IsValid) return View(model);

var existingBook = \_bookService.GetAllBooks().FirstOrDefault(b => b.Id == model.Id);

if (existingBook == null) return NotFound();

existingBook.Title = model.Title;

existingBook.Author = model.Author;

existingBook.ISBN = model.ISBN;

existingBook.Genre = model.Genre;

existingBook.Year = model.Year;

TempData["SuccessMessage"] = "Книга успешно обновлена!";

return RedirectToAction(nameof(Index));

}

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Create(BookViewModel model)

{

if (!ModelState.IsValid)

return View(model);

var book = new Book

{

Title = model.Title,

Author = model.Author,

ISBN = model.ISBN,

Genre = model.Genre,

Year = DateTime.Now.Year

};

\_bookService.AddBook(book);

Console.WriteLine($"Добавлена книга: {book.Title}, ID: {book.Id}");

TempData["SuccessMessage"] = "Книга успешно добавлена!";

return RedirectToAction(nameof(Index));

}

}

}

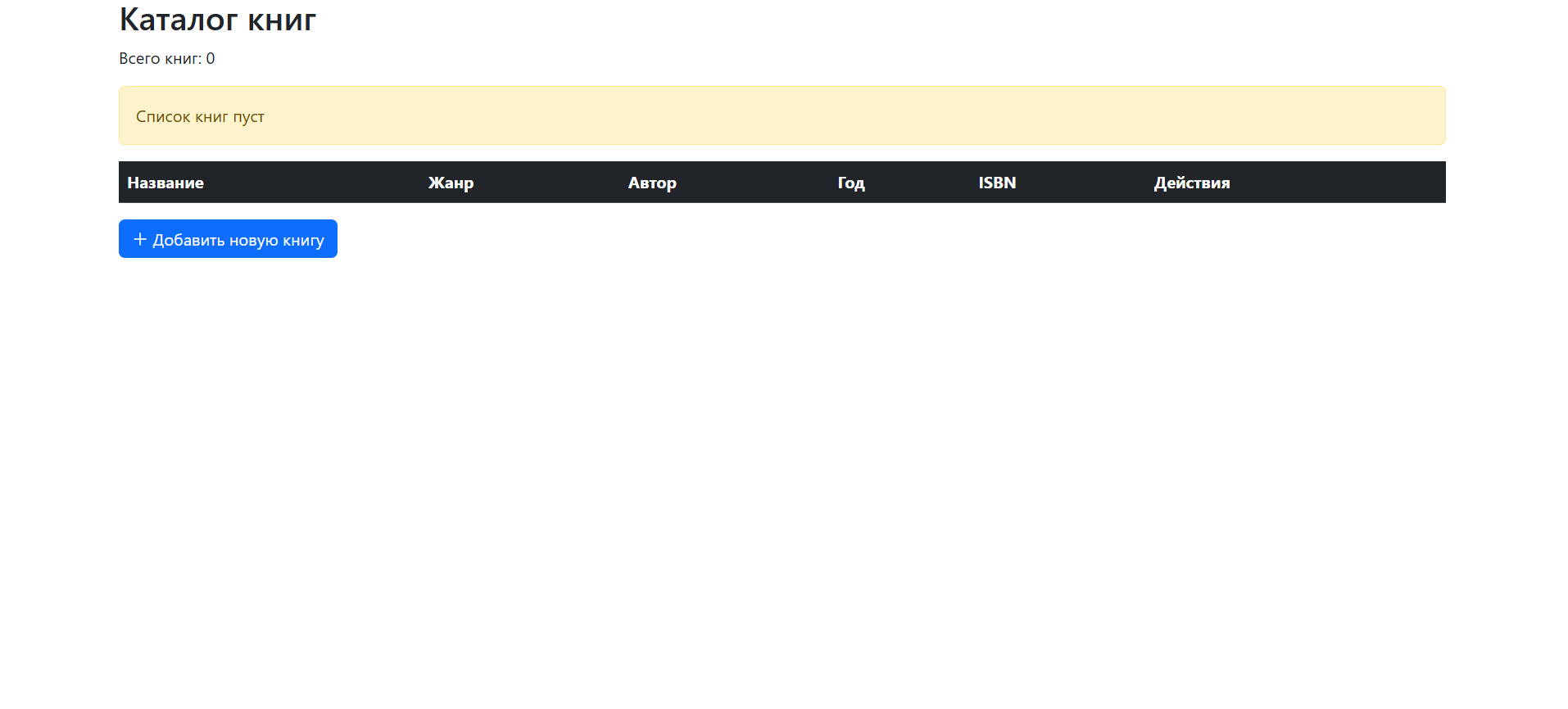


Рисунок 14.1 - Результат работы программы