CONTENTS

Abstract

As the number of books keeps on updating in one's book collection, it is very much difficult to keep tracks of the book. It would be much more convenient if we had some digital tools to manage the books we have. In this project, I created a website in which we can easily add the details of the book, update, delete and search the books easily. This project is named as **"Book Tracker"**. This project is considered as a simple database of book recording where the user can store books' records easily as it is not time-consuming. The website is linked to a SQL local database and the data can be managed much more easily with more security. The user has to input the name of the book, name of the author and ISBN number as the book details to create the book item.

Since, this project is in the beginning phase, there are many flaws in the website. On further days, I am planning to add many functionalities like authentication, views, better UI design etc. if this project gets good response from the user.

Introduction

Everybody has some books; some people may have a lot. As the number of books grows in one's collection, it would be convenient to have a software utility to manage them: generate reports on the books currently in collection, keep track of book loans, quickly find the books he or she needs from the bookshelves, etc. Also, many books now include their electronic version in CD-ROMs (normally in PDF or CHM format), and some authors even offer free download of their books. People also need an efficient way to manage these electronic documents, since the file names of these documents are often meaningless and therefore it becomes extremely hard to find the exact one needed in a collection of hundreds or thousands of electronic documents.

Problem Statement

The proposed book management system is a web application called **"Book Tracker"**. The goal of this project is to build a full-featured, commercial-quality software package to help people manage their books (either printed or electronic). Many people are having trouble to get exact details of the book they have. This application helps user to keep track of the books they have used before or they currently have and this software can be accessed easily by the user anytime. If they have a large number of books and if they want to get the details of particular book, they can easy search and retrieve the info of the book within seconds.

Objectives

The aims and objectives that will be achieved after completion of this project are as follows:

1. Register and login as a user.
2. Display all the books of your collection.
3. Add button to add the book. The parameters to add book includes book name, author name and ISBN number.
4. Edit button to update the details of the book.
5. Delete button to delete the book from your collection.

Study of Existing System

1. Song, Shanpeng, "A book management system eLibrary" (2004). Theses Digitization Project. 31. <https://scholarworks.lib.csusb.edu/etd-project/31>
2. <https://www.vidyarthiplus.com/vp/attachment.php?aid=24328>
3. DotNetSpider. 2005. We will design – you develop it https://www.dotnetspider.com/projects/7-Library-Management-System.aspx

Tools Used:

1. HTML, CSS, Bootstrap
2. SQL
3. ASP .NET

Introduction To Template Inheritance

Template inheritance is an approach to managing templates that resembles object-oriented programming techniques. Instead of the traditional use of {include ...} tags to manage parts of templates, you can inherit the contents of one template to another (like extending a class) and change blocks of content therein (like overriding methods of a class.) This keeps template management minimal and efficient, since each template only contains the differences from the template it extends.

Template inheritance allows us to build a base “skeleton” template that contains all the common elements of your site and defines blocks that child templates can override.

Introduction To Controller

MVC controllers are responsible for responding to requests made against an ASP.NET core MVC website. Each browser request is mapped to a particular controller. For example, imagine that you enter the following URL into the address bar of your browser:

http://localhost/Product/Index/3

In this case, a controller named ProductController is invoked. The ProductController is responsible for generating the response to the browser request. For example, the controller might return a particular view back to the browser or the controller might redirect the user to another controller.

Listing1 - Controllers\ProductController.cs

```
Using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Mvc.Ajax;

namespace MvcApplication1.Controllers
{ public class ProductController : Controller
     {
          // GET: /Products/

          public ActionResult Index()
          {
               // Add action logic here return
               View();
          }
     }
}
```

Razor Pages:

Register.cshtml

```
@page
@model RegisterModel
@{
    ViewData["Title"] = "Register";
}

<h1>@ViewData["Title"]</h1>

<div class="row">
    <div class="col-md-4">
        <form id="registerForm" asp-route-returnUrl="@Model.ReturnUrl" method="post">
            <h2>Create a new account.</h2>
            <hr />
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-floating">
                <input asp-for="Input.Email" class="form-control" autocomplete="username" aria-required="true" />
                <label asp-for="Input.Email"></label>
                <span asp-validation-for="Input.Email" class="text-danger"></span>
            </div>
            <div class="form-floating">
                <input asp-for="Input.Password" class="form-control" autocomplete="new-password" aria-required="true" />
                <label asp-for="Input.Password"></label>
                <span asp-validation-for="Input.Password" class="text-danger"></span>
            </div>
            <div class="form-floating">
                <input asp-for="Input.ConfirmPassword" class="form-control" autocomplete="new-password" aria-required="true"
/>
                <label asp-for="Input.ConfirmPassword"></label>
                <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
            </div>
            <button id="registerSubmit" type="submit" class="w-100 btn btn-lg btn-primary">Register</button>
        </form>
    </div>
    <div class="col-md-6 col-md-offset-2">
        <section>
            <h3>Use another service to register.</h3>
            <hr />
            @{
                if ((Model.ExternalLogins?.Count ?? 0) == 0)
                {
                    <div>
                        <p>
                            There are no external authentication services configured. See this <a
href="https://go.microsoft.com/fwlink/?LinkID=532715">article
                            about setting up this ASP.NET application to support logging in via external services</a>.
                        </p>
                    </div>
                }
                else
                {
                    <form id="external-account" asp-page="./ExternalLogin" asp-route-returnUrl="@Model.ReturnUrl"
method="post" class="form-horizontal">
                        <div>
                            <p>
                                @foreach (var provider in Model.ExternalLogins)
                                {
                                    <button type="submit" class="btn btn-primary" name="provider" value="@provider.Name" title="Log in
using your @provider.DisplayName account">@provider.DisplayName</button>
                                }
                            </p>
                </div>
            </form>
        }
    }
    </section>
  </div>
</div>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

Login.cshtml

```
@page
@model LoginModel

@{
    ViewData["Title"] = "Log in";
}

<h1>@ViewData["Title"]</h1>
<div class="row">
    <div class="col-md-4">
        <section>
            <form id="account" method="post">
                <h2>Use a local account to log in.</h2>
                <hr />
                <div asp-validation-summary="ModelOnly" class="text-danger"></div>
                <div class="form-floating">
                    <input asp-for="Input.Email" class="form-control" autocomplete="username" aria-required="true" />
                    <label asp-for="Input.Email" class="form-label"></label>
                    <span asp-validation-for="Input.Email" class="text-danger"></span>
                </div>
                <div class="form-floating">
                    <input asp-for="Input.Password" class="form-control" autocomplete="current-password" aria-required="true" />
                    <label asp-for="Input.Password" class="form-label"></label>
                    <span asp-validation-for="Input.Password" class="text-danger"></span>
                </div>
                <div>
                    <div class="checkbox">
                        <label asp-for="Input.RememberMe" class="form-label">
                            <input class="form-check-input" asp-for="Input.RememberMe" />
                            @Html.DisplayNameFor(m => m.Input.RememberMe)
                        </label>
                    </div>
                </div>
                <div>
                    <button id="login-submit" type="submit" class="w-100 btn btn-lg btn-primary">Log in</button>
                </div>
                <div>
                    <p>
                        <a id="forgot-password" asp-page="./ForgotPassword">Forgot your password?</a>
                    </p>
                    <p>
                        <a asp-page="./Register" asp-route-returnUrl="@Model.ReturnUrl">Register as a new user</a>
                    </p>
                    <p>
                        <a id="resend-confirmation" asp-page="./ResendEmailConfirmation">Resend email confirmation</a>
                    </p>
                </div>
            </form>
        </section>
    </div>
    <div class="col-md-6 col-md-offset-2">
        <section>
            <h3>Use another service to log in.</h3>
            <hr />
            @{
                if ((Model.ExternalLogins?.Count ?? 0) == 0)
                {
                    <div>
                        <p>
                            There are no external authentication services configured. See this <a
href="https://go.microsoft.com/fwlink/?LinkID=532715">article
                            about setting up this ASP.NET application to support logging in via external services</a>.
                        </p>
                    </div>
                }
                else
                {
                    <form id="external-account" asp-page="./ExternalLogin" asp-route-returnUrl="@Model.ReturnUrl" method="post"
class="form-horizontal">
                        <div>
                            <p>
                                @foreach (var provider in Model.ExternalLogins)
                                {
                                    <button type="submit" class="btn btn-primary" name="provider" value="@provider.Name" title="Log in
using your @provider.DisplayName account">@provider.DisplayName</button>
                                }
                            </p>
                        </div>
                    </form>
                }
            }
        </section>
    </div>
</div>

@section Scripts {
    <partial name="_ValidationScriptsPartial" />
}
```

Home/Index.cshtml

```
@{
    ViewData["Title"] = "Home Page";
}

<div class="text-center">
    <h1 class="display-4">Welcome To BookTracker</h1>
    <img src="img/homeimage.jpg" class="my-3"/>
    <br/>
    <button type="button" class="btn btn-outline-primary btn-lg">
        <a class="gotobookButton" asp-controller="Book" asp-action="Index">Go To Book List</a>
    </button>
</div>
```

Contact.cshtml

```
<div class="container contact-form">
            <div class="contact-image">
                <img src="https://image.ibb.co/kUagtU/rocket_contact.png" alt="rocket_contact"/>
            </div>
            <form method="post">
                <h3>Drop Us a Message</h3>
            <div class="row">
                <div class="col-md-6">
                    <div class="form-group mb-3">
                        <input type="text" name="txtName" class="form-control"                          placeholder="Your Name *"
value="" />
                    </div>
                    <div class="form-group mb-3">
                        <input type="text" name="txtEmail" class="form-control"   placeholder="Your Email *" value="" />
                    </div>
                    <div class="form-group mb-3">
                        <input type="text" name="txtPhone" class="form-control" placeholder="Your Phone Number *" value="" />
                    </div>
                    <div class="form-group">
                        <input type="submit" name="btnSubmit" class="btnContact" value="Send Message" />
                    </div>
                </div>
                <div class="col-md-6">
                    <div class="form-group">
                        <textarea name="txtMsg" class="form-control" placeholder="Your Message *" style="width: 100%; height:
150px;"></textarea>
                    </div>
                </div>
            </div>
        </form>
</div>
```

Book/Index.cshtml

```
@model IEnumerable<Book>
@{
    ViewData["Title"] = "Index";
}

<div class="container row p-0 m-0">
    <div class="col-9">
        <h2 class="text-info">Book List</h2>
    </div>
    <div class="col-3">
        <a asp-controller="Book" asp-action="Create" class="btn btn-info form-control text-white">Create New Book</a>
    </div>

    <div class="col-12 border p-3 mt-3">
        <form method="post">
            @if (Model.Count() > 0)
            {
                <table class="table table-striped border">
                    <tr class="table-secondary">
                        <th>
                            Book Name
                        </th>
                        <th>
                            Author
                        </th>
                        <th>
                            ISBN No.
                        </th>
                        <th>
                            Actions
                        </th>
                    </tr>
                    @foreach(var item in Model)
                    {
                        <tr>
                            <td>
                                @item.Name
                            </td>
                            <td>
                                @item.Author
                            </td>
                            <td>
                                @item.ISBN
                            </td>
                            <td>
                                <button asp-controller="Book" asp-route-Id="@item.Id" asp-action="Edit" class="btn btn-success btn-sm
text-white">Edit</button>
                                <button asp-controller="Book" asp-route-Id="@item.Id" asp-action="Delete"  class="btn btn-danger btn-
sm">Delete</button>
                            </td>
                        </tr>
                    }
                </table>
            }
            else
            {
                <p>No books available.</p>
            }
        </form>
    </div>
</div>
```

Create.cshtml

```
@using BookTrackerMVC.Areas.Identity.Data
@using Microsoft.AspNetCore.Identity
@model Book
@inject UserManager<AuthUser> UserManager

<h2 class="text-info">Create New Book</h2>

<div class="border container" style="padding:30px;">
    <form method="post" asp-action="Create">
        <input type="hidden" asp-for="userId" value="@UserManager.GetUserId(User)"/>
        <div class="text-danger" asp-validation-summary="ModelOnly"></div>
        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Name"></label>
            </div>
            <div class="col-6">
                <input asp-for="Name" class="form-control"/>
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Author"></label>
            </div>
            <div class="col-6">
                <input asp-for="Author" class="form-control"/>
                <span asp-validation-for="Author" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="ISBN"></label>
            </div>
            <div class="col-6">
                <input asp-for="ISBN" class="form-control"/>
                <span asp-validation-for="ISBN" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row">
            <div class="col-3 offset-3">
                <input type="submit" value="Create" class="btn btn-primary form-control"/>
            </div>
            <div class="col-3">
                <a asp-action="Index" class="btn btn-success form-control">Back to List</a>
            </div>
        </div>
    </form>
</div>

@section Scripts{
    @{<partial name="_ValidationScriptsPartial.cshtml"/>}
}
```

Edit.cshtml

```
@model Book

<h2 class="text-info">Update Book</h2>

<div class="border container" style="padding:30px;">
    <form method="post" asp-controller="Book" asp-action="EditPost">
        <input type="hidden" asp-for="Id" />
        <input type="hidden" asp-for="userId"  />
        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Name"></label>
            </div>
            <div class="col-6">
                <input asp-for="Name" class="form-control"/>
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Author"></label>
            </div>
            <div class="col-6">
                <input asp-for="Author" class="form-control"/>
                <span asp-validation-for="Author" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="ISBN"></label>
            </div>
            <div class="col-6">
                <input asp-for="ISBN" class="form-control"/>
                <span asp-validation-for="ISBN" class="text-danger"></span>
            </div>
        </div>

        <div class="form-group row">
            <div class="col-3 offset-3">
                <input type="submit" value="Update" class="btn btn-warning form-control"/>
            </div>
            <div class="col-3">
                <a asp-action="Index" class="btn btn-success form-control">Back to List</a>
            </div>
        </div>
    </form>
</div>

@section Scripts{
    @{<partial name="_ValidationScriptsPartial.cshtml"/>}
}
```

Delete.cshtml

```
@model Book

<h2 class="text-info">Delete Book</h2>

<div class="border container" style="padding:30px;">
    <form method="post" asp-action="DeletePost">
        <input type="hidden" asp-for="Id" />
        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Name"></label>
            </div>
            <div class="col-6">
                <input asp-for="Name" disabled class="form-control"/>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="Author"></label>
            </div>
            <div class="col-6">
                <input asp-for="Author" disabled class="form-control"/>
            </div>
        </div>

        <div class="form-group row mb-3">
            <div class="col-3">
                <label asp-for="ISBN"></label>
            </div>
            <div class="col-6">
                <input asp-for="ISBN" disabled class="form-control"/>
            </div>
        </div>

        <div class="form-group row">
            <div class="col-3 offset-3">
                <input type="submit" value="Delete" class="btn btn-danger form-control"/>
            </div>
            <div class="col-3">
                <a asp-action="Index" class="btn btn-success form-control">Back to List</a>
            </div>
        </div>
    </form>
</div>

@section Scripts{
    @{<partial name="_ValidationScriptsPartial.cshtml"/>}
}
```

Controller Pages

BookController.cs

```
using BookTrackerMVC.Areas.Identity.Data;
using BookTrackerMVC.Data;
using BookTrackerMVC.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Identity;
using Microsoft.AspNetCore.Mvc;

namespace BookTrackerMVC.Controllers
{
    [Authorize]
    public class BookController : Controller
    {

        private readonly ApplicationDbContext _db;
        public BookTrackerMVCContext context;
        public readonly UserManager<AuthUser> userManager;

        public BookController(ApplicationDbContext _db, BookTrackerMVCContext context, UserManager<AuthUser> userManager
)
        {
            this._db = _db;
            this.context = context;
            this.userManager = userManager;
        }
        public IActionResult Index()
        {
            IEnumerable<Book> objList = _db.Book;
            var filteredList = objList.Where(item => item.userId == userManager.GetUserId(User));
            return View(filteredList);
        }

        public IActionResult Create()
        {
            return View();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> Create(Book obj)
        {
                _db.Book.Add(obj);
                await _db.SaveChangesAsync();
                return RedirectToAction("Index");
        }

        public IActionResult Edit(int? id)
        {
            if(id == null || id == 0)
            {
                return NotFound();
            }
            var obj = _db.Book.Find(id);
            if(obj == null)
            {
                return NotFound();
            }
            return View(obj);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<IActionResult> EditPost(Book obj)
        {
            _db.Book.Update(obj);
            await _db.SaveChangesAsync();
            return RedirectToAction("Index");
        }

        public IActionResult Delete(int? id)
        {
            if (id == null || id == 0)
            {
                return NotFound();
            }
            var obj = _db.Book.Find(id);
            if (obj == null)
            {
                return NotFound();
            }
            return View(obj);
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public IActionResult DeletePost(int? id)
        {
            var obj = _db.Book.Find(id);
            if (obj == null)
            {
                return NotFound();
            }
            _db.Book.Remove(obj);
            _db.SaveChanges();
            return RedirectToAction("Index");
        }

        public IActionResult SignUp()
        {
            return View();
        }

        public IActionResult Login()
        {
            return View();
        }
    }
}
```

HomeController.cs

```
using BookTrackerMVC.Models;
using Microsoft.AspNetCore.Mvc;
using System.Diagnostics;

namespace BookTrackerMVC.Controllers
{
    public class HomeController : Controller
    {
        private readonly ILogger<HomeController> _logger;

        public HomeController(ILogger<HomeController> logger)
        {
            _logger = logger;
        }

        public IActionResult Index()
        {
            return View();
        }

        public IActionResult Privacy()
        {
            return View();
        }

        public IActionResult Contact()
        {
            return View();
        }

        [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]
        public IActionResult Error()
        {
            return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });
        }
    }
}
```

Other Pages

appsettings.json

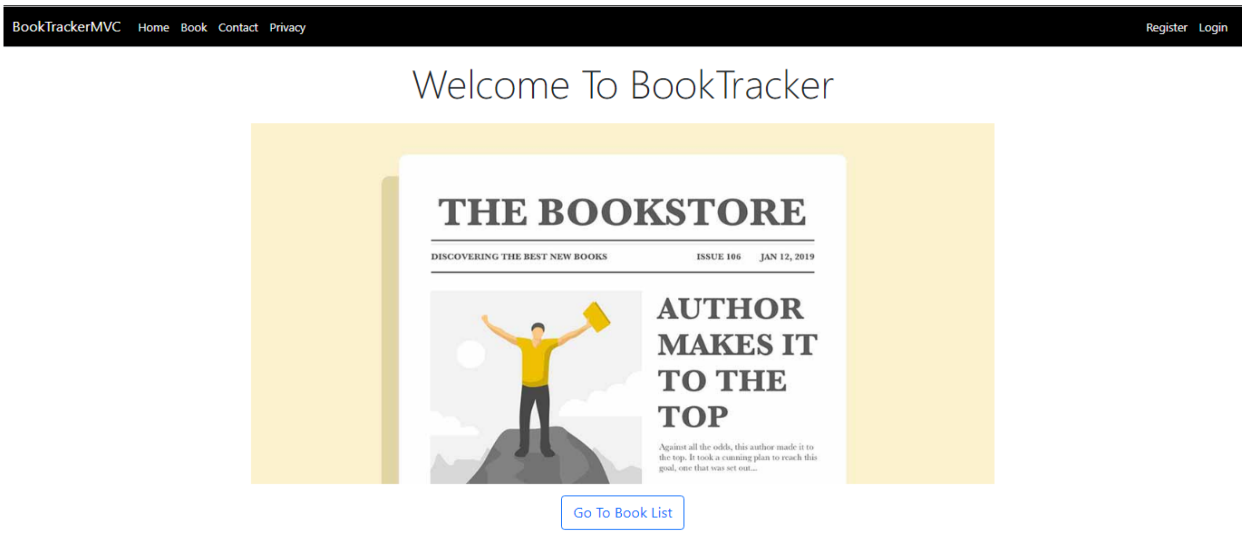
```
{
  "ConnectionStrings": {
    "DefaultConnection":
"Server=localhost\\MSSQLSERVER01;Database=BookListMVC;Trusted_Connection=True;MultipleActiveResultSets=True",
    "BookTrackerMVCContextConnection":
"Server=localhost\\MSSQLSERVER01;Database=BookListMVC;Trusted_Connection=True;MultipleActiveResultSets=True"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

Program.cs

```
using BookTrackerMVC.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.AspNetCore.Identity;
using BookTrackerMVC.Areas.Identity.Data;

var builder = WebApplication.CreateBuilder(args);
var connectionString = builder.Configuration.GetConnectionString("BookTrackerMVCContextConnection") ?? throw new
InvalidOperationException("Connection string 'BookTrackerMVCContextConnection' not found.");

builder.Services.AddDbContext<BookTrackerMVCContext>(options =>
    options.UseSqlServer(connectionString));;

builder.Services.AddDefaultIdentity<AuthUser>(options => options.SignIn.RequireConfirmedAccount = true)
    .AddEntityFrameworkStores<BookTrackerMVCContext>();;

// Add services to the container.
builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(
    builder.Configuration.GetConnectionString("DefaultConnection")
    ));
builder.Services.AddControllersWithViews().AddRazorRuntimeCompilation();
var app = builder.Build();

// Configure the HTTP request pipeline.
if (!app.Environment.IsDevelopment())
{
    app.UseExceptionHandler("/Home/Error");.
    app.UseHsts();
}

app.UseHttpsRedirection();
app.UseStaticFiles();
app.UseRouting();
app.UseAuthentication();;
app.UseAuthorization();
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
app.MapRazorPages();
app.Run();
```

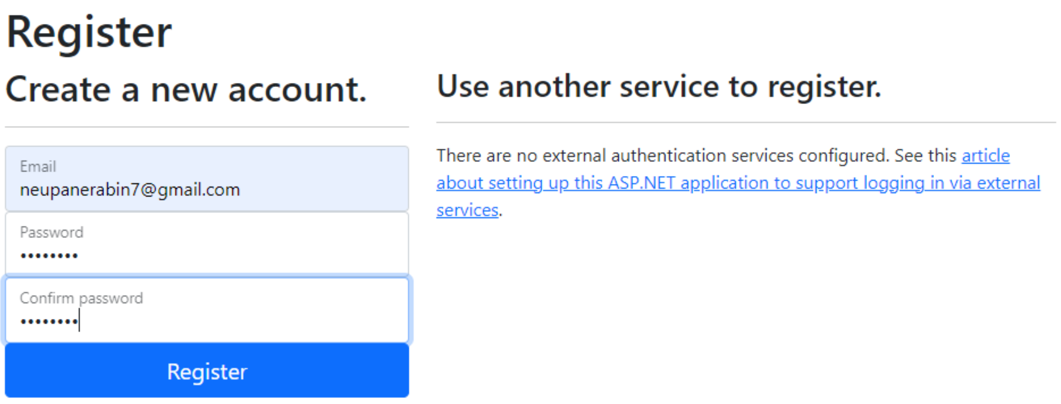
Output:

In Home Page:



Welcome To BookTracker

In Register page:



Register

Create a new account.

Email
neupanerabin7@gmail.com

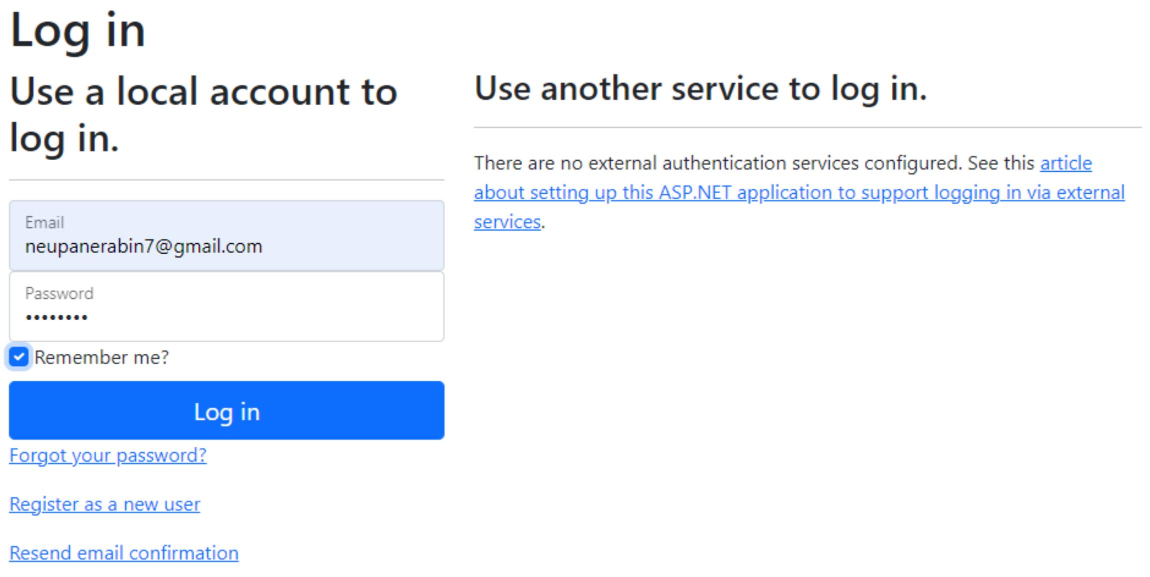
Password
••••••••

Confirm password
••••••••

Use another service to register.

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

In Login page:



Log in

Use a local account to log in.

Email
neupanerabin7@gmail.com

Password
••••••••

☑ Remember me?

Log in

Forgot your password?

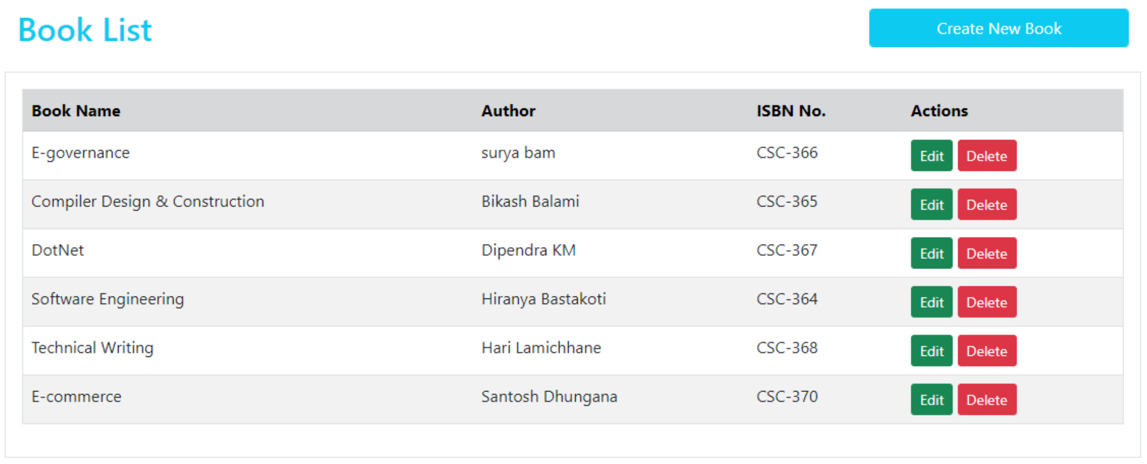
Register as a new user

Resend email confirmation

Use another service to log in.

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

In book/home page:

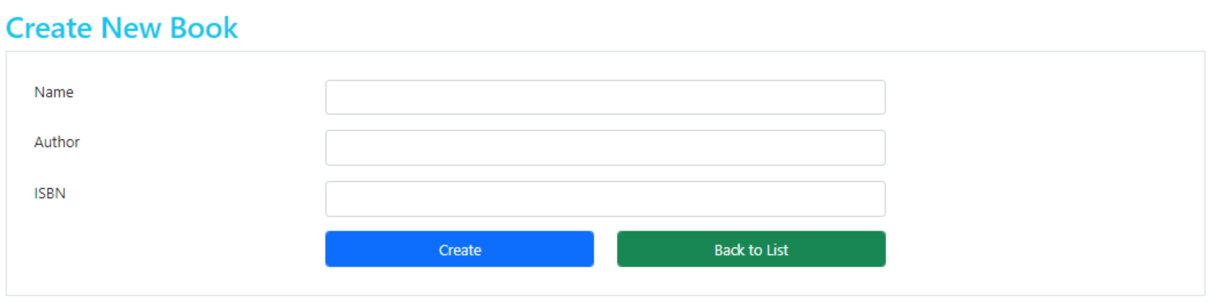


Book List

Create New Book

Book Name	Author	ISBN No.	Actions
E-governance	surya bam	CSC-366	Edit Delete
Compiler Design & Construction	Bikash Balami	CSC-365	Edit Delete
DotNet	Dipendra KM	CSC-367	Edit Delete
Software Engineering	Hiranya Bastakoti	CSC-364	Edit Delete
Technical Writing	Hari Lamichhane	CSC-368	Edit Delete
E-commerce	Santosh Dhungana	CSC-370	Edit Delete

In Create page:

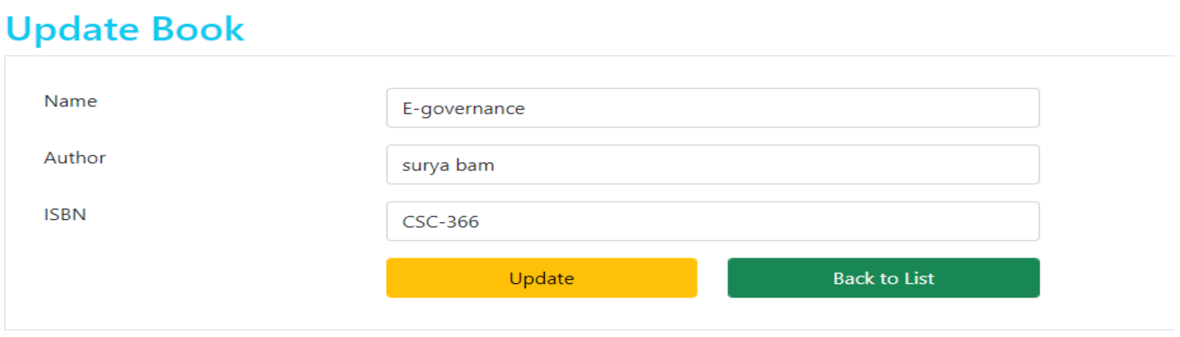


Create New Book

Name	
Author	
ISBN	

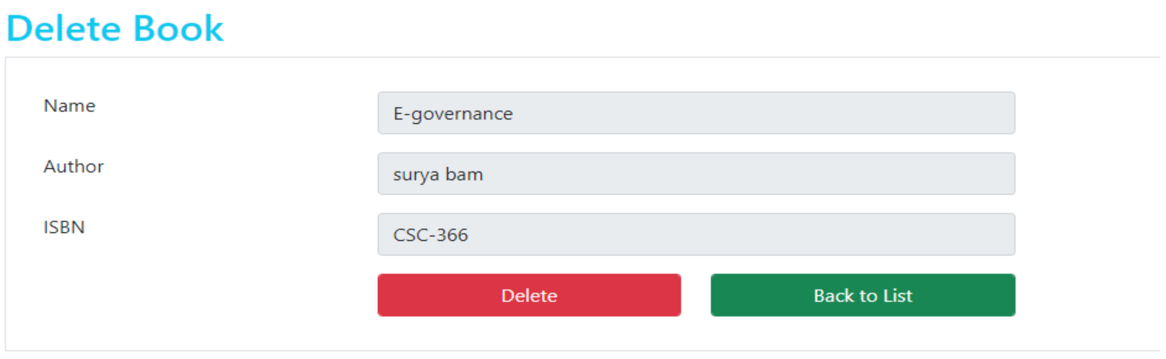
Create | Back to List

In Edit page:


Update Book

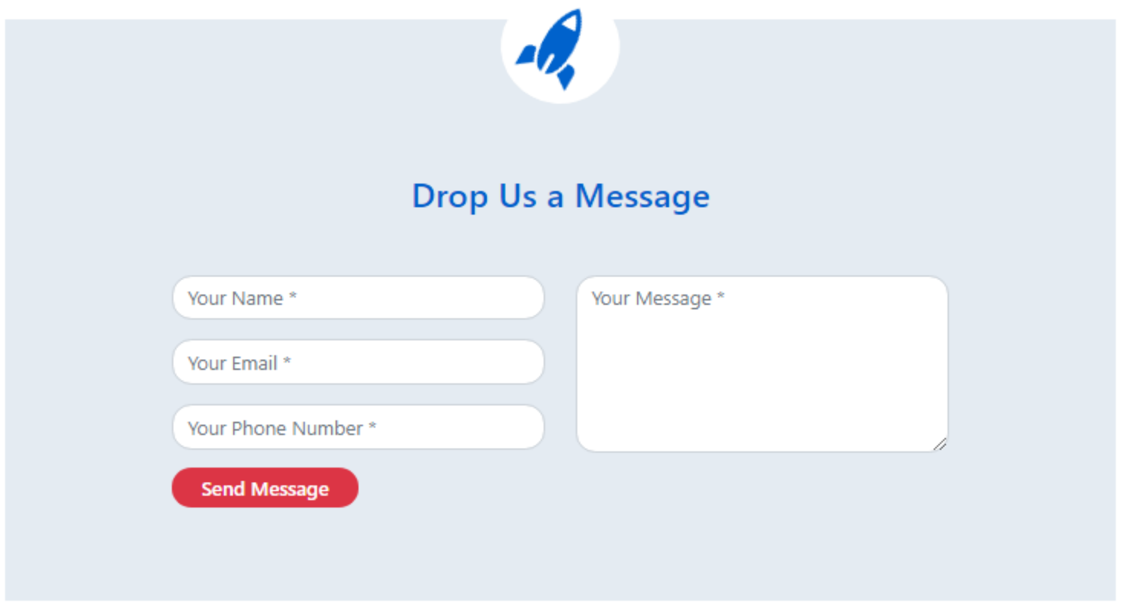
Name	E-governance
Author	surya bam
ISBN	CSC-366

In Delete page:


Delete Book

Name	E-governance
Author	surya bam
ISBN	CSC-366

In Contact page:


Drop Us a Message

Conclusion:

To conclude, this research is done to analyze how MVC plays an important role in Object Oriented Software development. An amazing architectural design pattern i.e. Model-View-Controller design pattern helps to maintain and define the good quality of website. By using MVC in object-oriented web development can make software more flexible, clear, reliable and scalable.