

20.12.2024

## Sieć Web i chmura (4)



Opracowywanie zawartości dla platformy ASP.NET i sie... ☒

Twórz aplikacje internetowe dla wielu platform przy użyciu technologii ASP.NET Core, ASP.NET, HTML/JavaScript i ko...

mvc

Wyczyść wszystko

Wszystkie języki

Wszystkie platformy

Wszystkie typy projektów



Aplikacja internetowa ASP.NET Core (Model-View-Controller)

Szablon projektu służący do tworzenia aplikacji platformy ASP.NET Core z przykładowymi widokami i kontrolerami platformy ASP.NET Core MVC. Tego szablonu można także użyć dla usług HTTP RESTful.

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

## Konfiguruj nowy projekt

Aplikacja internetowa ASP.NET Core (Model-View-Controller)

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

Nazwa projektu

AplikacjaTrenowania

Lokalizacja

F:\

Nazwa rozwiązania ⓘ

AplikacjaTrenowania

☐ Umieść rozwiązanie i projekt w tym samym katalogu

Projekt utworzymy w ścieżce „F:\AplikacjaTrenowania\AplikacjaTrenowania\”

## Informacje dodatkowe

Aplikacja internetowa ASP.NET Core (Model-View-Controller)

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

Platforma ⓘ

.NET 8.0 (Długoterminowa pomoc techniczna)

Typ uwierzytelniania ⓘ

Brak

☒ Konfiguruj dla protokołu HTTPS ⓘ

☐ Włączanie obsługi kontenerów ⓘ

System operacyjny kontenera ⓘ

Linux

Typ kompilacji kontenera ⓘ

Dockerfile

☐ Nie używaj instrukcji najwyższego poziomu ⓘ

☐ Rejestracja w orkiestracji .NET Aspire ⓘ

Wersja Aspire ⓘ

9.0

## Dodaj nowy element szkieletowy

▲ Zainstalowane

Tożsamość

Układ

▲ Wspólne

API

▸ Blazor

▸ MVC

Strony Razor



Tożsamość

**Tożsamość**

przez Microsoft  
v1.0.0.0

Dodaje kod wymagany do używania tożsamości platformy ASP.NET Core w aplikacji.

Identyfikator: IdentityScaffolder

Dodaj

Anuluj

## Dodaj Tożsamość

Wybierz istniejącą stronę układu lub określ nową:

/Areas/Identity/Pages/Account/Manage/\_Layout.cshtml

(Zostaw pustą, jeśli jest ustawiona w pliku Razor \_viewstart)

☐ Zastąp wszystkie pliki

Wybierz pliki do zastąpienia

☐ Account\StatusMessage

☐ Account\ConfirmEmailChange

☐ Account\ForgotPasswordConfirmation

☐ Account>LoginWith2fa

☐ Account\Manage\Layout

☐ Account\Manage\ChangePassword

☐ Account\Manage\DownloadPersonalData

☐ Account\Manage\ExternalLogins

☐ Account\Manage\PersonalData

☐ Account\Manage\ShowRecoveryCodes

☐ Account\RegisterConfirmation

☐ Account\ResetPasswordConfirmation

☐ Account\AccessDenied

☐ Account\ExternalLogin

☐ Account\Lockout

☐ Account>LoginWithRecoveryCode

☐ Account\Manage\ManageNav

☐ Account\Manage\DeletePersonalData

☐ Account\Manage\Email

☐ Account\Manage\GenerateRecoveryCodes

☐ Account\Manage\ResetAuthenticator

☐ Account\Manage\TwoFactorAuthentication

☐ Account\ResendEmailConfirmation

☐ Account\ConfirmEmail

☐ Account\ForgotPassword

☒ Account>Login

☒ Account\Logout

☐ Account\Manage\StatusMessage

☐ Account\Manage\Disable2fa

☐ Account\Manage\EnableAuthenticator

☐ Account\Manage\Index

☐ Account\Manage\SetPassword

☒ Account\Register

☐ Account\ResetPassword

Klasa DbContext

ApplicationDbContext



Dostawca bazy danych

SQL Server



Klasa użytkownika

ApplicationUser



Dodaj

Anuluj

# SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

## Server

Server type: Database Engine

Server name: DESKTOP-TFJR5RH

Authentication: Windows Authentication

User name: DESKTOP-TFJR5RH\Damian

Password:

☐ Remember password

## Connection Security

Encryption: Optional

☐ Trust server certificate

Host name in certificate:

Connect

Cancel

Help

Options <<

Support for ASP.NET Core Identity was added to your project.

**Microsoft.EntityFrameworkCore.SqlServer** przeź Microsoft  
Microsoft SQL Server database provider for Entity Framework Core.

8.0.11  
9.0.0

**Microsoft.EntityFrameworkCore** przeź Microsoft  
Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations. EF Core works with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.

8.0.11  
9.0.0

Tworzenie repozytorium Git

Wypchnij do nowego zdalnego

GitHub

Azure DevOps

Inne

Istniejące zdalne

Tylko lokalne

Zainicjuj lokalne repozytorium Git

Ścieżka lokalna

F:\AplikacjaTrenowania

Szablon .gitignore

Domyślne (VisualStudio)

Szablon licencji

Brak

☐ Dodaj README.md

Utwórz nowe repozytorium GitHub

Konto

DamianSikoraUP (GitHub)

Właściciel

DamianSikoraUP

Nazwa repozytorium

AplikacjaTrenowania

Opis

Wprowadź opis repozytorium GitHub <Opcjonalnie>

Widoczność

Private

Wybierz, kto może wyświetlać to repozytorium i zatwierdzać w nim.

Przekaż kod do usługi GitHub

<https://github.com/DamianSikoraUP/AplikacjaTrenowania>

Utwórz i wypchnij

Anuluj

General

Access

Collaborators

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Who has access

Private repository

Only those with access to this repository can view it

Manage

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

DIRECT ACCESS

1 user has access to this repository.

1 collaborator.

Manage access

Add people

Select all

Type

Find a collaborator...

DawidBilUP

Collaborator

Previous

Next

22.12.2024

Przygotowaliśmy początkowe szkice które miały za zadanie rozplanować działanie aplikacji

Aplikacja Treningowa | ~~Trening~~ | Woda | Białko | Login / Rejestry

DATA GODZINA

Rodzaj treningu LISTA

Wybierz ćwiczenia Lista

Kłosa 1.  kg /  powtórzeń

Dodaj Serię

Zapisz

Zapisz Ćwiczenie.

Zapisane :



Aplikacja Treningowa | Trening | Woda | Białko | Login / Rejestracja

DATA

Dodaj wypitą wodę (ml)

Zapisz

Wypita woda

----- ml

Dzienny cel wody

----- ml

Aplikacja Treningowa | Trening | Woda | Białko | Login / Rejestracja

DATA

Dobry Źródło białko (g)



Zapisz

Źródło białko

----- g

Dzienny cel białka

----- g

HomeController.cs\* [icon] X

AplikacjaTrenowania [icon] AplikacjaTrenowania.C

```
{
  using System.Diagnostics;
  using AplikacjaTrenowania.Models;
  using Microsoft.AspNetCore.Mvc;

  namespace AplikacjaTrenowania.Controllers
  {
    Odwołania: 3
    public class HomeController : Controller
    {
      private readonly ILogger<HomeController> _logger;

      Odwołania: 0
      public HomeController(ILogger<HomeController> logger)
      {
        _logger = logger;
      }

      Odwołania: 0
      public IActionResult Index()
      {
        return View();
      }

      Odwołania: 0
      public IActionResult Privacy()
      {
        return View();
      }

      Odwołania: 0
      public IActionResult Trening()
      {
        return View();
      }
    }
  }
```

Odwołania: 0

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

Odwołania: 0

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

- [icon] Zapytaj Copilot Alt+/
  - [icon] Dodaj widok...
  - [icon] Przejdź do widoku Ctrl+M, Ctrl+G
  - [icon] Szybkie akcje i refaktoryzacje... Ctrl+.
  - [icon] Zmień nazwę... Ctrl+R, Ctrl+R
  - Usuń i sortuj instrukcje Usings Ctrl+R, Ctrl+G
  - [icon] Wgląd w definicję Alt+F12
  - [icon] Przejdź do definicji F12



## Dodaj nowy element szkieletowy

### ▲ Zainstalowane

Tożsamość  
Układ

### ▲ Wspólne

API

▸ Blazor

### ▲ MVC

Kontroler

Wyświetl

Strony Razor



Widok Razor — pusty



Widok Razor

### Widok Razor

przez Microsoft  
v1.0.0.0

Widok Razor z silnie typizowanym  
modelem lub bez niego

Identyfikator: RazorViewScaffolder

Dodaj

Anuluj

## Dodaj Widok Razor

Nazwa widoku

Trening

Szablon

Empty (bez modelu)

Klasa modelu

Klasa DbContext

Dostawca bazy danych

### Opcje



Utwórz jako widok częściowy



Odwołaj się do bibliotek skryptów

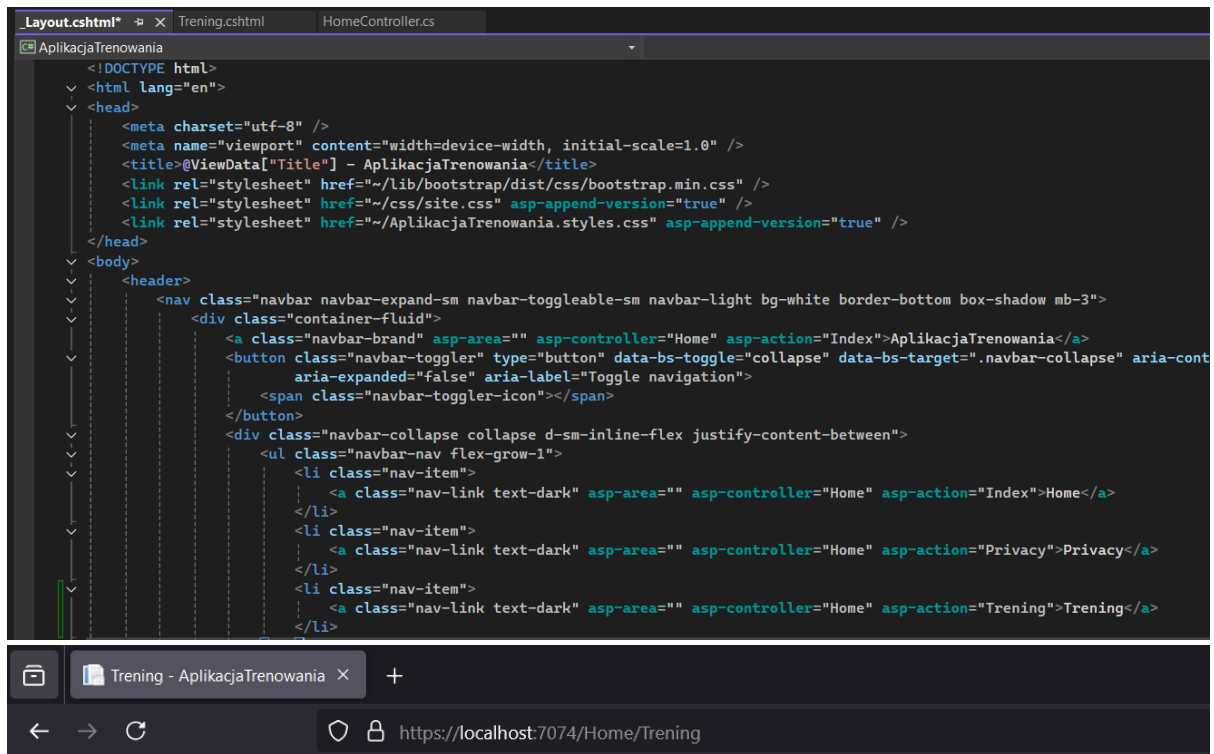


Użyj strony układu

(Zostaw pustą, jeśli jest ustawiona w pliku Razor \_viewstart)

Dodaj

Anuluj

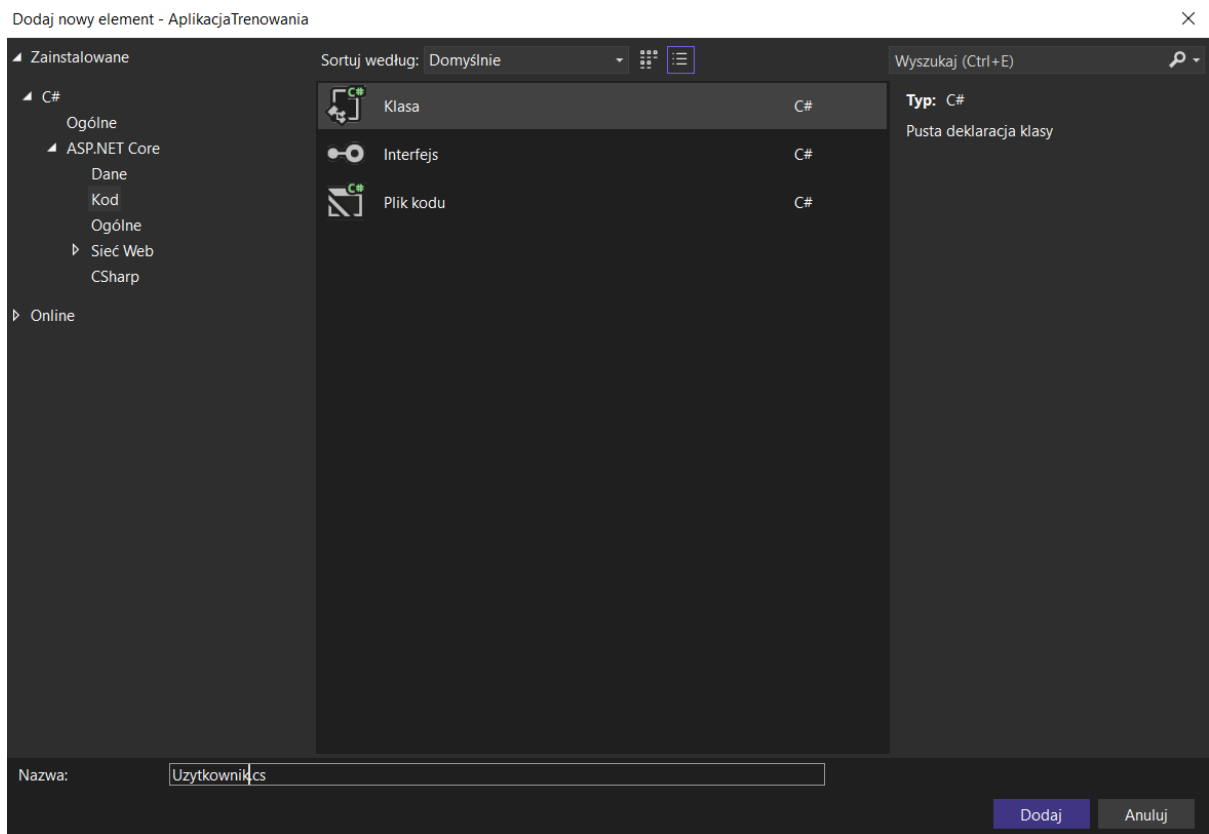
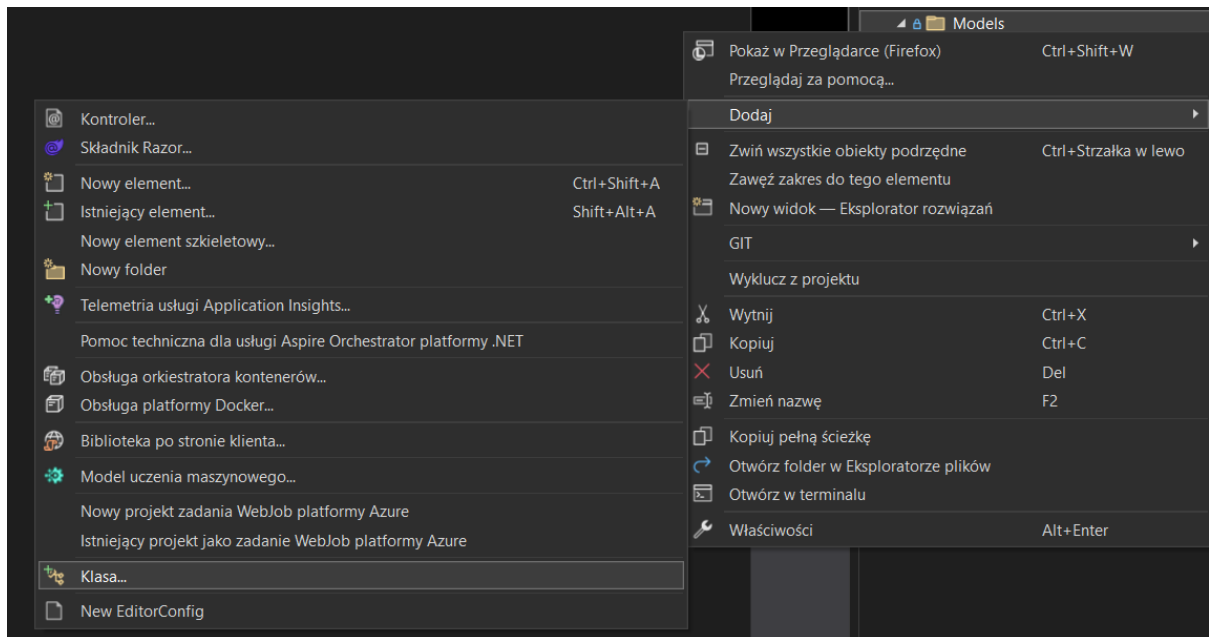


AplikacjaTrenowania   Home   Privacy   Trening

## Trening

Analogicznie dodaliśmy podstrony Woda i Białko

AplikacjaTrenowania   Home   Privacy   Trening   Woda   Białko



```
{ } namespace AplikacjaTrenowania.Models
{
    Odwołania: 0
    public class Uzytkownik
    {
        Odwołania: 0
        public int Id { get; set; }
        Odwołania: 0
        public required string Login { get; set; }
        Odwołania: 0
        public required string Email { get; set; }
        Odwołania: 0
        public int DziennyCelWody { get; set; }
        Odwołania: 0
        public int IleWypitoWody { get; set; }
        Odwołania: 0
        public int DziennyCelBialkowy { get; set; }
        Odwołania: 0
        public int IleZjedzonoBialka { get; set; }
    }
}
```

23.12.2024

▲ Zainstalowane

▲ C#

- Ogólne
- ▲ ASP.NET Core
  - Dane
  - Kod
  - Ogólne
  - Sieć Web
    - CSharp

- Online

Sortuj według: Domyślnie

- 
-

Wyszukaj (Ctrl+E)

	Klasa	C#	<b>Typ:</b> C# Pusta deklaracja klasy
	Interfejs	C#	
	Plik kodu	C#	

Nazwa:

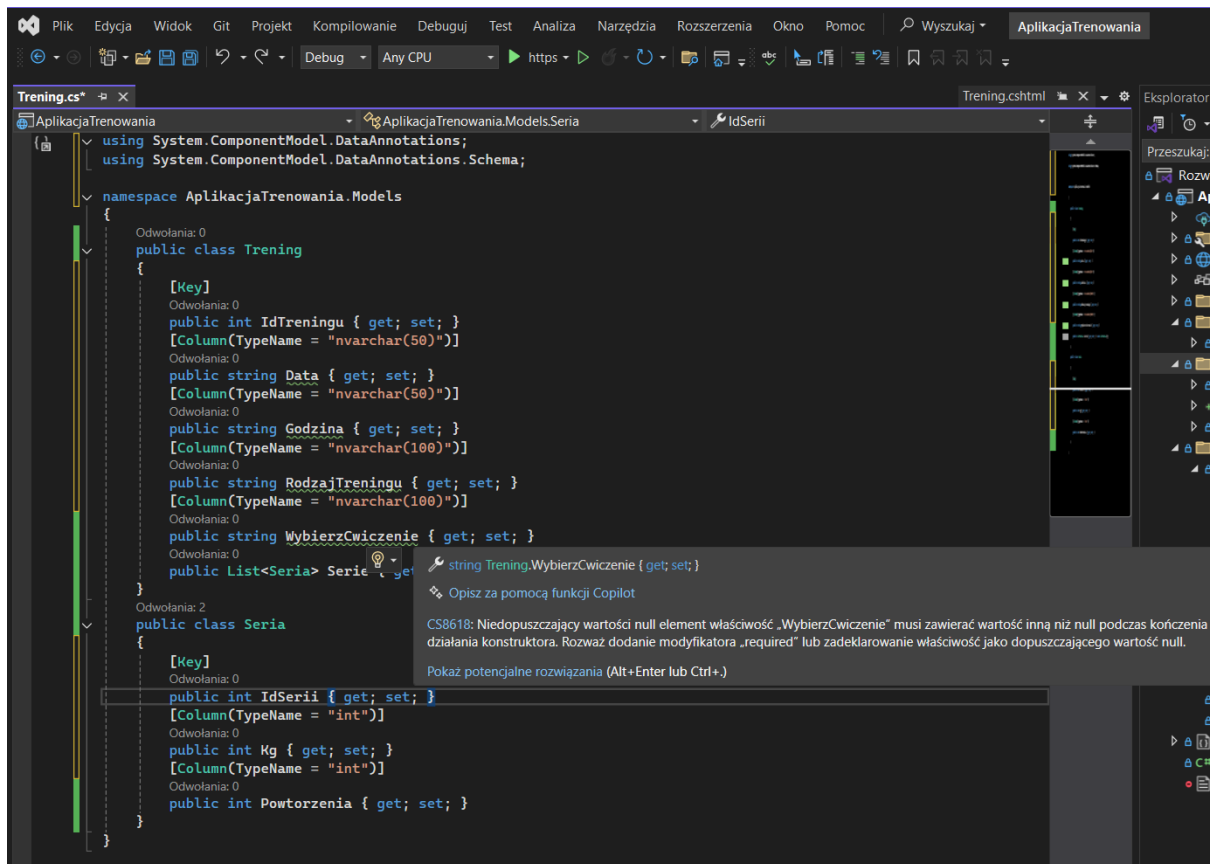
Dodaj

Anuluj

```
Trening.cs*
AplikacjaTrenowania
AplikacjaTrenowania.Models.Trening

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace AplikacjaTrenowania.Models
{
    Odwołania: 0
    public class Trening
    {
        [Key]
        Odwołania: 0
        public int IdTreningu { get; set; }
        [Column(TypeName = "nvarchar(50)")]
        Odwołania: 0
        public string Data { get; set; }
        [Column(TypeName = "nvarchar(50)")]
        Odwołania: 0
        public string Godzina { get; set; }
        [Column(TypeName = "nvarchar(100)")]
        Odwołania: 0
        public string RodzajTreningu { get; set; }
        [Column(TypeName = "nvarchar(100)")]
        Odwołania: 0
        public string WybierzCwiczenie { get; set; }
        Odwołania: 0
        public List<Seria> Serie { get; set; } = new List<Seria>();
    }
    Odwołania: 2
    public class Seria
    {
        [Key]
        Odwołania: 0
        public int IdSerii { get; set; }
        [Column(TypeName = "int")]
        Odwołania: 0
        public int Kg { get; set; }
        [Column(TypeName = "int")]
        Odwołania: 0
        public int Powtorzenia { get; set; }
    }
}
```

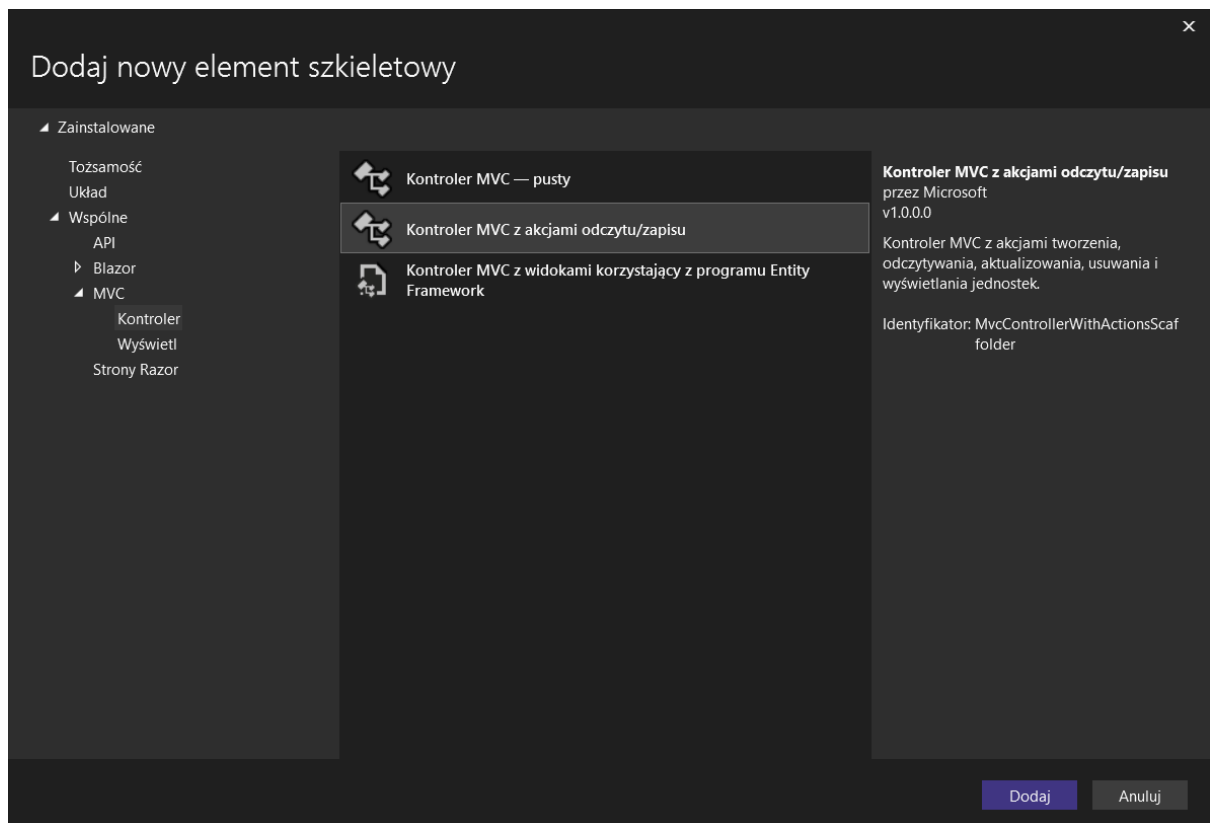


```
Trening.cs*
AplikacjaTrenowania
AplikacjaTrenowania.Models.Trening
WybierzCwiczenie

using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace AplikacjaTrenowania.Models
{
    Odwołania: 0
    public class Trening
    {
        [Key]
        Odwołania: 0
        public int IdTreningu { get; set; }
        [Column(TypeName = "nvarchar(50)")]
        Odwołania: 0
        public string Data { get; set; } = string.Empty;
        [Column(TypeName = "nvarchar(50)")]
        Odwołania: 0
        public string Godzina { get; set; } = string.Empty;
        [Column(TypeName = "nvarchar(100)")]
        Odwołania: 0
        public string RodzajTreningu { get; set; } = string.Empty;
        [Column(TypeName = "nvarchar(100)")]
        Odwołania: 0
        public string WybierzCwiczenie { get; set; } = string.Empty;
        Odwołania: 0
        public List<Seria> Serie { get; set; } = new List<Seria>();
    }
    Odwołania: 2
    public class Seria
    {
        [Key]
        Odwołania: 0
        public int IdSerii { get; set; }
        [Column(TypeName = "int")]
        Odwołania: 0
        public int Kg { get; set; }
        [Column(TypeName = "int")]
        Odwołania: 0
        public int Powtorzenia { get; set; }
    }
}
```





Usuujemy stary widok trening

## Dodaj nowy element szkieletowy

### ◀ Zainstalowane

Tożsamość

Układ

#### ◀ Wspólne

API

▸ Blazor

#### ◀ MVC

Kontroler

Wyświetl

Strony Razor



Widok Razor — pusty



Widok Razor

#### Widok Razor

przez Microsoft  
v1.0.0.0

Widok Razor z silnie typizowanym modelem lub bez niego

Identyfikator: RazorViewScaffolder

Dodaj

Anuluj

## Dodaj Widok Razor

Nazwa widoku

Index

Szablon

List

Klasa modelu

Trening (AplikacjaTrenowania.Models)

Klasa DbContext

<Required>

Dostawca bazy danych

<Required>

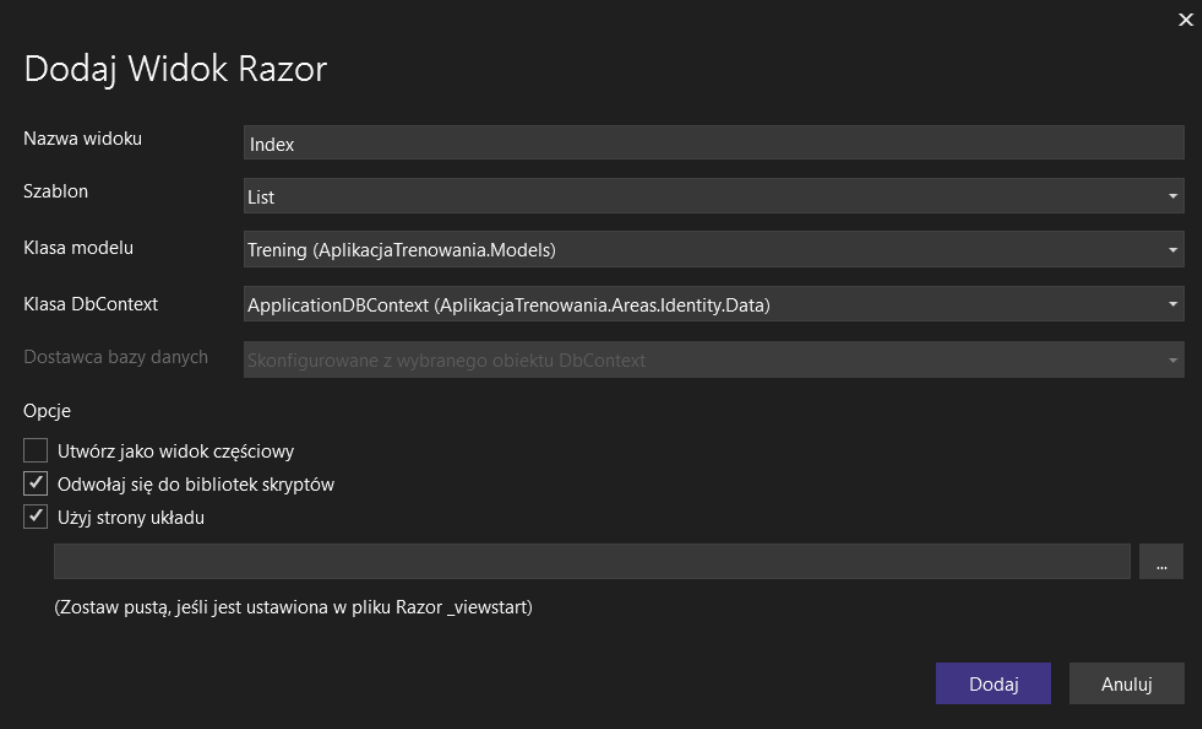
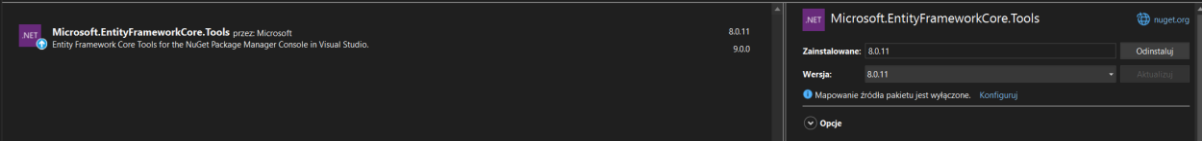
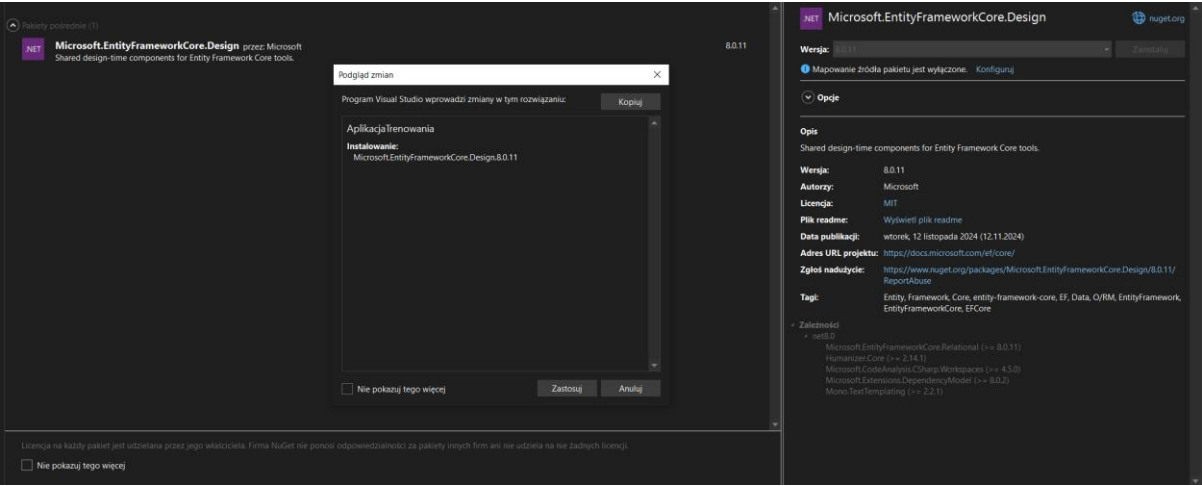
### Opcje

- ☐ Utwórz jako widok częściowy
- ☒ Odwołaj się do bibliotek skryptów
- ☒ Użyj strony układu

(Zostaw pustą, jeśli jest ustawiona w pliku Razor \_viewstart)

Dodaj

Anuluj



Dodaj Widok Razor

Nazwa widoku

Create

Szablon

Create

Klasa modelu

Trening (AplikacjaTrenowania.Models)

Klasa DbContext

ApplicationDbContext (AplikacjaTrenowania.Areas.Identity.Data)

Dostawca bazy danych

Skonfigurowane z wybranego obiektu DbContext

Opcje

☐ Utwórz jako widok częściowy

☒ Odwołaj się do bibliotek skryptów

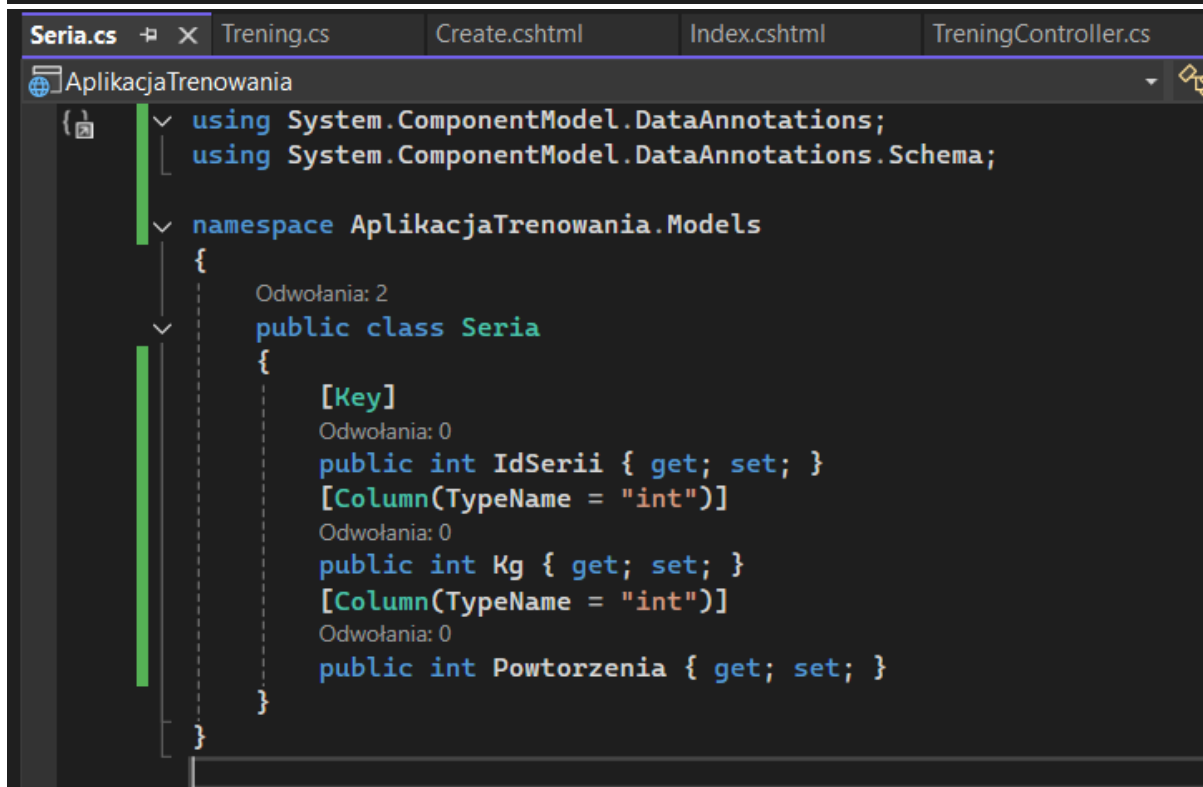
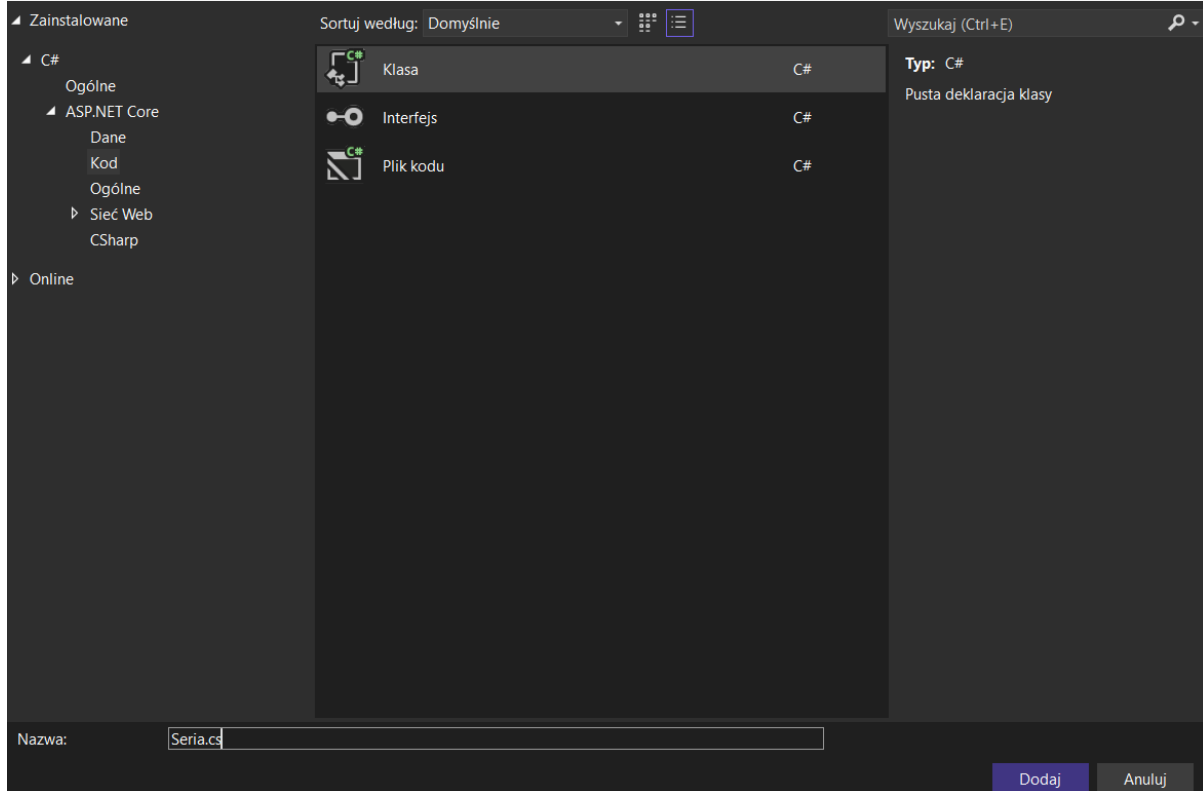
☒ Użyj strony układu

(Zostaw pustą, jeśli jest ustawiona w pliku Razor \_viewstart)

Dodaj

Anuluj

Popelniliśmy błąd, mieliśmy 2 klasy w jednym pliku i przez to klasa seria nie została użyta. 2 Klasy mieliśmy dlatego, że wtedy nam działało tworzenie listy ( a przynajmniej myślimy że działało), ale teraz musimy znaleźć na to inny sposób





```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml  Trening.cs  TreningController.cs
AplikacjaTrenowania  AplikacjaTrenowania.Models.Trening

1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace AplikacjaTrenowania.Models
5  {
6      Odwołania: 11
7      public class Trening
8      {
9          [Key]
10         Odwołania: 3
11         public int IdTreningu { get; set; }
12         [Column(TypeName = "nvarchar(50)")]
13         Odwołania: 5
14         public string Data { get; set; } = DateTime.Now.Date.ToString("d");
15         [Column(TypeName = "nvarchar(50)")]
16         Odwołania: 5
17         public string Godzina { get; set; } = DateTime.Now.ToString("HH:mm");
18         [Column(TypeName = "nvarchar(100)")]
19         Odwołania: 4
20         public string RodzajTreningu { get; set; } = string.Empty;
21         [Column(TypeName = "nvarchar(100)")]
22         Odwołania: 4
23         public string WybierzCwiczenie { get; set; } = string.Empty;
24         1 odwołanie
25         public List<Seria> Serie { get; set; } = new List<Seria>();
26     }
27 }
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml  Trening.cs  TreningController.cs
AplikacjaTrenowania  AplikacjaTrenowania.Controllers.TreningController

1  using Microsoft.AspNetCore.Http;
2  using Microsoft.AspNetCore.Mvc;
3  using AplikacjaTrenowania.Models;
4
5  namespace AplikacjaTrenowania.Controllers
6  {
7      Odwołania: 3
8      public class TreningController : Controller
9      {
10         private readonly ILogger<TreningController> _logger;
11         private static List<Trening> zapisaneTreningi = new List<Trening>();
12         private readonly Dictionary<string, List<string>> cwiczenia = new Dictionary<string, List<string>>
13         {
14             { "Siłowy", new List<string> { "Przysiady", "Martwy ciąg", "Wyciskanie sztangi" } },
15             { "Kardio", new List<string> { "Bieganie", "Rower", "Skakanka" } },
16             { "Stretching", new List<string> { "Joga", "Rozciąganie dynamiczne", "Pilates" } }
17         };
18         Odwołania: 0
19         public TreningController(ILogger<TreningController> logger)
20         {
21             _logger = logger;
22         }
23         // GET: TreningController
24         1 odwołanie
25         public ActionResult Index()
26         {
27             return View();
28         }
29
30         // GET: TreningController/Details/5
31         Odwołania: 0
32         public ActionResult Details(int id)
33         {
34             return View();
35         }
36
37         // GET: TreningController/Create
38         Odwołania: 0
39         public ActionResult Create()
40         {
41             ViewBag.Cwiczenia = cwiczenia;
42             return View(new Trening());
43         }
44
45         // POST: TreningController/Create
46         [HttpPost]
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml  Trening.cs  TreningController.cs
AplikacjaTrenowania  AplikacjaTrenowania.Controllers.TreningController

40      // POST: TreningController/Create
41      [HttpPost]
42      [ValidateAntiForgeryToken]
43      Odwołania: 0
44      public ActionResult Create(IFormCollection collection)
45      {
46          try
47          {
48              return RedirectToAction(nameof(Index));
49          }
50          catch
51          {
52              return View();
53          }
54      }
55      Odwołania: 0
56      public IActionResult ZapiszTrening(Trening model, int[] serieKg, int[] seriePowtorzenia)
57      {
58          for (int i = 0; i < serieKg.Length; i++)
59          {
60              model.Serie.Add(new Seria
61              {
62                  Kg = serieKg[i],
63                  Powtorzenia = seriePowtorzenia[i]
64              });
65          }
66          zapisaneTreningi.Add(model);
67          return RedirectToAction("ListaTreningow");
68      }
69  }
70  }
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTrenowania

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>@ViewData["Title"] - AplikacjaTrenowania</title>
7      <link rel="stylesheet" href="~/Lib/bootstrap/dist/css/bootstrap.min.css" />
8      <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
9      <link rel="stylesheet" href="~/AplikacjaTrenowania.styles.css" asp-append-version="true" />
10 </head>
11 <body>
12     <header>
13         <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
14             <div class="container-fluid">
15                 <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AplikacjaTrenowania</a>
16                 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-controls="navbarSupportedContent"
17                     aria-expanded="false" aria-label="Toggle navigation">
18                     <span class="navbar-toggler-icon"></span>
19                 </button>
20                 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
21                     <ul class="navbar-nav flex-grow-1">
22                         <li class="nav-item">
23                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Index">Trening</a>
24                         </li>
25                         <li class="nav-item">
26                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Woda">Woda</a>
27                         </li>
28                         <li class="nav-item">
29                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Biatko">Biatko</a>
30                         </li>
31                     </ul>
32                 </div>
33             </div>
34         </nav>
35     </header>
36     <div class="container">
37         <main role="main" class="pb-3">
38             @RenderBody()
39         </main>
40     </div>
41     <footer class="border-top footer text-muted">
42         <div class="container">
43             &copy; 2024 - AplikacjaTrenowania - <a asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
44         </div>
45 </body>
46 </html>
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTrenowania

47 <script src="~/Lib/jquery/dist/jquery.min.js"></script>
48 <script src="~/Lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
49 <script src="~/js/site.js" asp-append-version="true"></script>
50 @await RenderSectionAsync("Scripts", required: false)
51 </body>
52 </html>
53 </html>
```



```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTreningowa
1  @model AplikacjaTreningowa.Models.Trening
2
3  @{
4      ViewData["Title"] = "Create";
5  }
6
7  <h1>Create</h1>
8
9  <h4>Trening</h4>
10 <hr />
11 <div class="row">
12     <div class="col-md-4">
13         <form asp-action="Create">
14             <div asp-validation-summary="ModelOnly" class="text-danger"></div>
15             <div class="form-group">
16                 <label asp-for="Data" class="control-label"></label>
17                 <input asp-for="Data" class="form-control" readonly />
18                 <span asp-validation-for="Data" class="text-danger"></span>
19             </div>
20             <div class="form-group">
21                 <label asp-for="Godzina" class="control-label"></label>
22                 <input asp-for="Godzina" class="form-control" readonly />
23                 <span asp-validation-for="Godzina" class="text-danger"></span>
24             </div>
25             <div class="form-group">
26                 <label asp-for="RodzajTreningu" class="control-label"></label>
27                 <select id="rodzajTreningu" asp-for="RodzajTreningu" class="form-control">
28                     <option value="" disabled selected>Wybierz rodzaj treningu</option>
29                     @foreach (var rodzaj in ViewBag.Cwiczenia.Keys)
30                     {
31                         <option value="@rodzaj">@rodzaj</option>
32                     }
33                 </select>
34             </div>
35             <div class="form-group">
36                 <label asp-for="WybierzCwiczenie" class="control-label"></label>
37                 <select id="wybierzCwiczenie" asp-for="WybierzCwiczenie" class="form-control">
38                     <option value="" disabled selected>Wybierz ćwiczenie</option>
39                     @foreach (var item in ViewBag.Cwiczenia)
40                     {
41                         @foreach (var cwiczenie in item.Value)
42                         {
43                             <option data-cwiczenie="@item.Key">@cwiczenie</option>
44                         }
45                     }
46                 </select>
47             </div>
48             <div id="form-group-seria">
49                 <div class="row">
50                     <div class="seryjka">
51                         <input type="number" placeholder="Kg" class="form-control seria-kg" />
52                     </div>
53                     <div class="seryjka">
54                         <input type="number" placeholder="Powtórzeń" class="form-control seria-powtorzenia" />
55                     </div>
56                     <div class="seryjka">
57                         <button type="button" class="btn btn-danger remove-seria">Usuń</button>
58                     </div>
59                 </div>
60                 <button type="button" id="addSeria" class="btn btn-primary mt-3">Dodaj serię</button>
61                 <button type="submit" class="btn btn-success mt-3">Zapisz Trening</button>
62             </div>
63         </form>
64     </div>
65 </div>
66
67 <div>
68     <a asp-action="Index">Back to List</a>
69 </div>
70
71 <script src="~/js/form.js"></script>
72 <script src="~/js/addserie.js"></script>
73
74 @section Scripts {
75     @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
76 }
77
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTreningowa
44     }
45     }
46     </select>
47 </div>
48 <div id="form-group-seria">
49     <div class="row">
50         <div class="seryjka">
51             <input type="number" placeholder="Kg" class="form-control seria-kg" />
52         </div>
53         <div class="seryjka">
54             <input type="number" placeholder="Powtórzeń" class="form-control seria-powtorzenia" />
55         </div>
56         <div class="seryjka">
57             <button type="button" class="btn btn-danger remove-seria">Usuń</button>
58         </div>
59     </div>
60     <button type="button" id="addSeria" class="btn btn-primary mt-3">Dodaj serię</button>
61     <button type="submit" class="btn btn-success mt-3">Zapisz Trening</button>
62 </div>
63 </form>
64 </div>
65 </div>
66
67 <div>
68     <a asp-action="Index">Back to List</a>
69 </div>
70
71 <script src="~/js/form.js"></script>
72 <script src="~/js/addserie.js"></script>
73
74 @section Scripts {
75     @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
76 }
77
```

# AplikacjaTrenowania

```

1  @model IEnumerable<AplikacjaTrenowania.Models.Trening>
2
3  @{
4      ViewData["Title"] = "Index";
5  }
6
7  <h1>Index</h1>
8
9  <p>
10     <a asp-action="Create">Create New</a>
11 </p>
12 <table class="table">
13     <thead>
14         <tr>
15             <th>
16                 @Html.DisplayNameFor(model => model.Data)
17             </th>
18             <th>
19                 @Html.DisplayNameFor(model => model.Godzina)
20             </th>
21             <th>
22                 @Html.DisplayNameFor(model => model.RodzajTreningu)
23             </th>
24             <th>
25                 @Html.DisplayNameFor(model => model.WybierzCwiczenie)
26             </th>
27         </tr>
28     </thead>
29     <tbody>
30
31     @if (Model != null)
32     {
33         @foreach (var item in Model) {
34             <tr>
35                 <td>
36                     @Html.DisplayFor(modelItem => item.Data)
37                 </td>
38                 <td>
39                     @Html.DisplayFor(modelItem => item.Godzina)
40                 </td>
41                 <td>
42                     @Html.DisplayFor(modelItem => item.RodzajTreningu)
43                 </td>
44                 <td>
45                     @Html.DisplayFor(modelItem => item.WybierzCwiczenie)

```

formjs X addseriejs Program.cs Index.cshtml X Create.cshtml \_Layout.cshtml\* Trenig.cs TrenigController.cs

# AplikacjaTrenowania

```

45     @Html.DisplayFor(modelItem => item.WybierzCwiczenie)
46 </td>
47 <td>
48     <a asp-action="Edit" asp-route-id="@item.IdTreningu">Edit</a> |
49     <a asp-action="Details" asp-route-id="@item.IdTreningu">Details</a> |
50     <a asp-action="Delete" asp-route-id="@item.IdTreningu">Delete</a>
51 </td>
52 </tr>
53 }
54 </tbody>
55 </table>
56
57

```

# AplikacjaTrenowania

```

1  using Microsoft.AspNetCore.Identity;
2  using Microsoft.EntityFrameworkCore;
3  using AplikacjaTrenowania.Areas.Identity.Data;
4  var builder = WebApplication.CreateBuilder(args);
5  var connectionString = builder.Configuration.GetConnectionString("ApplicationDbContextConnection") ?? throw new InvalidOperationException("Connection string 'ApplicationDbContextConnection' not found.");
6
7  builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer(connectionString));
8
9  builder.Services.AddDefaultIdentity<ApplicationUser>(options => options.SignIn.RequireConfirmedAccount = true).AddEntityFrameworkStores<ApplicationDbContext>();
10
11 // Add services to the container.
12 builder.Services.AddControllersWithViews();
13
14 var app = builder.Build();
15
16 // Configure the HTTP request pipeline.
17 if (app.Environment.IsDevelopment())
18 {
19     app.UseExceptionHandler("/Home/Error");
20     // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
21     app.UseHsts();
22 }
23
24 app.UseHttpsRedirection();
25 app.UseStaticFiles();
26
27 app.UseRouting();
28
29 app.UseAuthorization();
30
31 app.MapControllerRoute(
32     name: "default",
33     pattern: "{controller=Home}/{action=Index}/{id?}");
34
35 app.Run();
36

```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTreningowa Pliki zawartości języka JavaScript  <global>
1  document.getElementById('addSerie').addEventListener('click', function () {
2      console.log('Dodaj serie');
3      const serieContainer = document.getElementById('form-group-seria');
4      const seria = document.createElement('div');
5      seria.classList.add('row', 'mt-2');
6
7      seria.innerHTML = `
8          <div class="seryjka">
9              <input type="number" placeholder="kg" class="form-control seria-kg" />
10             </div>
11             <div class="seryjka">
12                 <input type="number" placeholder="Powtórzenia" class="form-control seria-powtorzenia" />
13             </div>
14             <div class="seryjka">
15                 <button type="button" class="btn btn-danger remove-seria">Usuń</button>
16             </div>
17         `;
18     serieContainer.appendChild(serie);
19
20     // Dodaj obsługę usuwania serii
21     seria.querySelector('.remove-seria').addEventListener('click', function () {
22         seria.remove();
23     });
24 }
```

```
form.js  addserie.js  Program.cs  Index.cshtml  Create.cshtml  _Layout.cshtml*  Trening.cs  TreningController.cs
AplikacjaTreningowa Pliki zawartości języka JavaScript  <global>
1  const select1 = document.querySelector("#rodzajTreningu");
2  const select2 = document.querySelector("#wybierzCwiczenie");
3  const options = select2.querySelectorAll("option");
4
5  select1.addEventListener("change", () => {
6
7      options.forEach(option => {
8          option.hidden = true;
9          option.disabled = true;
10      });
11
12      const trening = select1.value;
13
14      options.forEach(option => {
15          if (option.dataset.cwiczenie === trening) {
16              option.hidden = false;
17              option.disabled = false;
18          }
19      });
20
21      select2.selectedIndex = -1;
22  });
```

# Create

## Trening

Data

03.01.2025

Godzina

17:08

RodzajTreningu

Wybierz rodzaj treningu

WybierzCwiczenie

Wybierz ćwiczenie

Kg

Powtórzenia

Usuń

Dodaj serię

Zapisz Trening

[Back to List](#)

# Create

## Trening

Data

03.01.2025

Godzina

17:08

RodzajTreningu

Silowy

WybierzCwiczenie

Przysiady

100

8

Usuń

100

8

Usuń

100

6

Usuń

Dodaj serię

Zapisz Trening

[Back to List](#)