

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 x Wyczyszczyć 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 Usluga Internet

Konfiguruj nowy projekt

Aplikacja internetowa ASP.NET Core (Model-View-Controller) C# Linux macOS Windows Chmura Usluga 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 Usluga Internet

Platforma ⓘ
.NET 8.0 (Długoterminowa pomoc techniczna)

Typ uwierzytelniania ⓘ
Brak

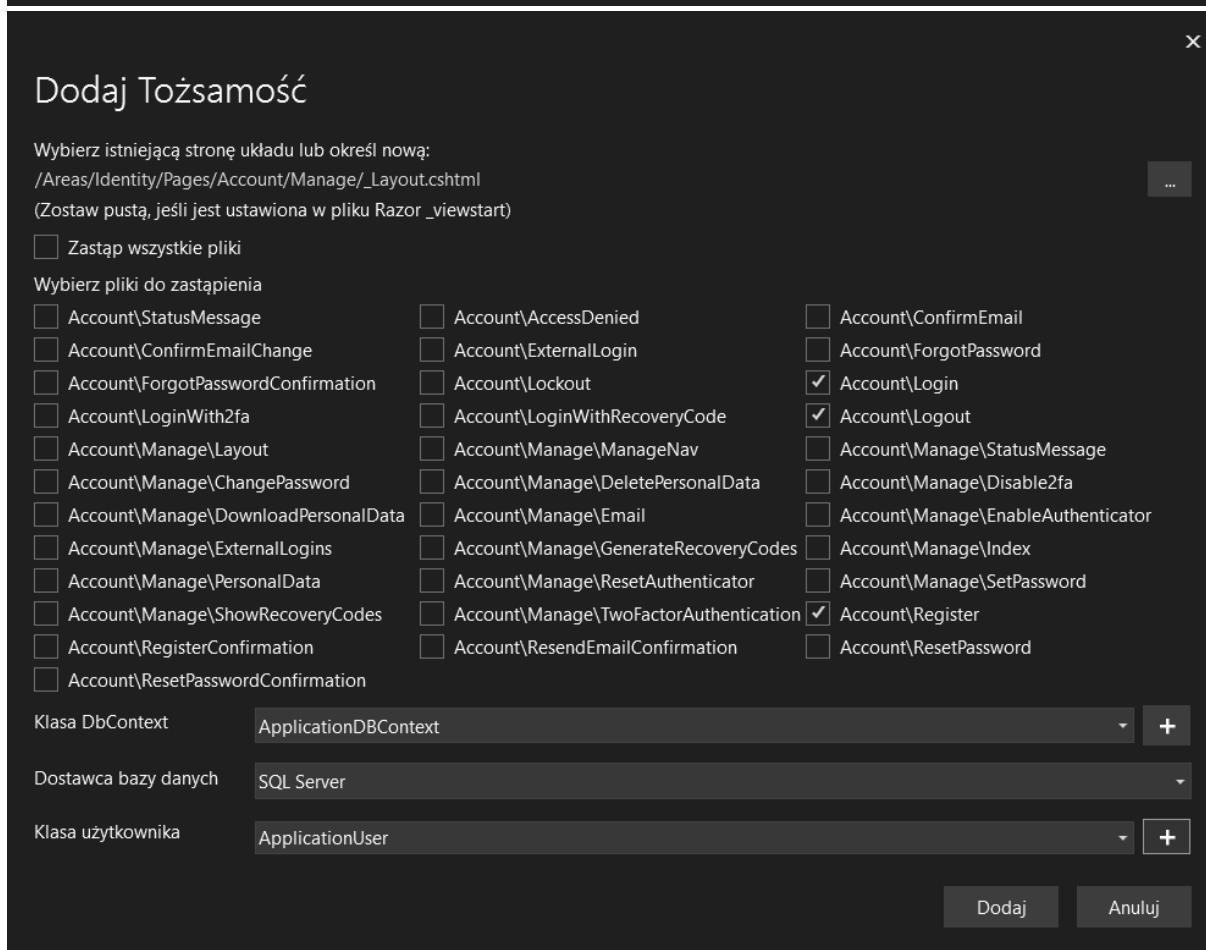
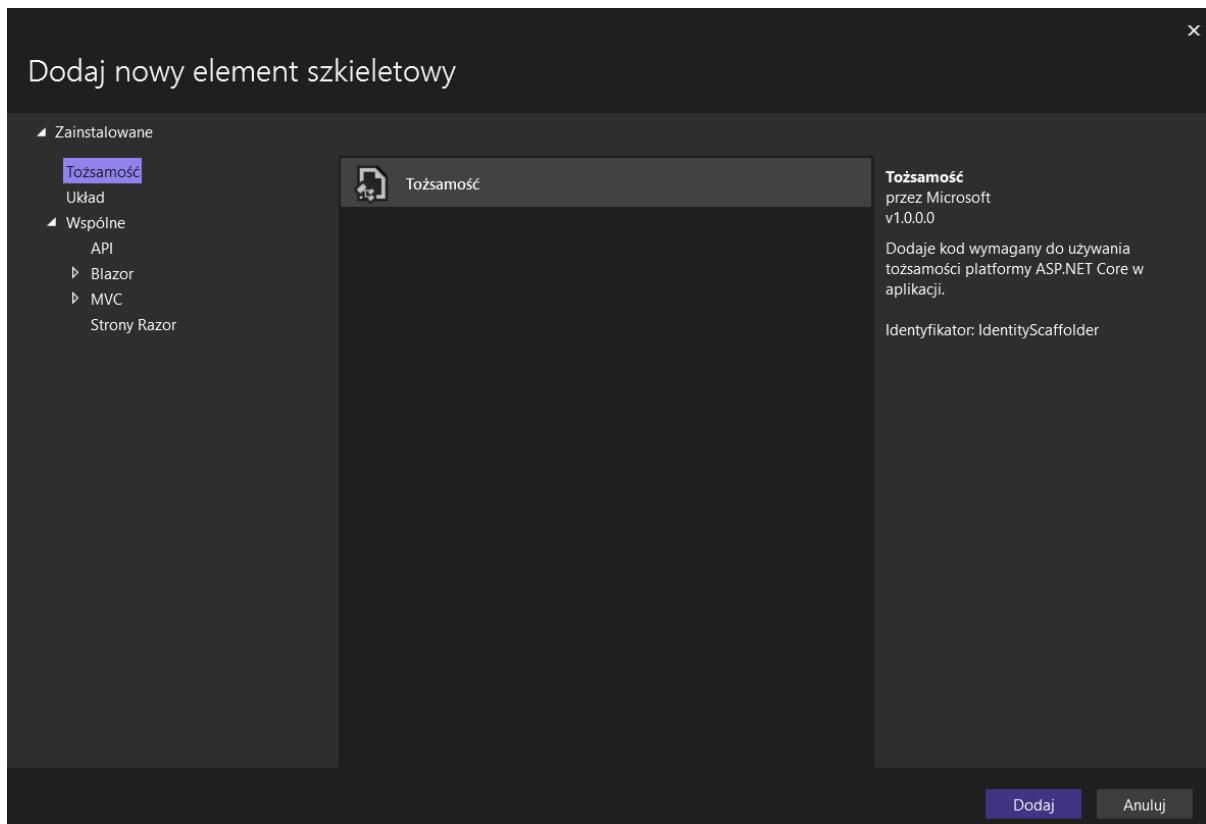
Konfiguruj dla protokołu HTTPS ⓘ
 Włączanie obsługi kontenerów ⓘ

System operacyjny kontenera ⓘ
Linux

Typ komplikacji kontenera ⓘ
Dockerfile

Nie używaj instrukcji najwyższego poziomu ⓘ
 Rejestracja w orkiestracji .NET Aspire ⓘ

Wersja Aspire ⓘ
9.0



Connect to Server X

SQL Server

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.

.NET Microsoft.EntityFrameworkCore.SqlServer przez: Microsoft
Microsoft SQL Server database provider for Entity Framework Core. 8.0.11 9.0.0

.NET Microsoft.EntityFrameworkCore przez: 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**

Who has access

Private repository
Only those with access to this repository can view it **Manage**

Code and automation

- Branches
- Tags
- Rules
- Actions
- Webhooks
- Codespaces
- Pages

Security

- Code security
- Deploy keys
- Secrets and variables

Integrations

- GitHub Apps
- Email notifications

DIRECT ACCESS
1 user has access to this repository.
[1 collaborator.](#) **Manage**

Manage access **Add people**

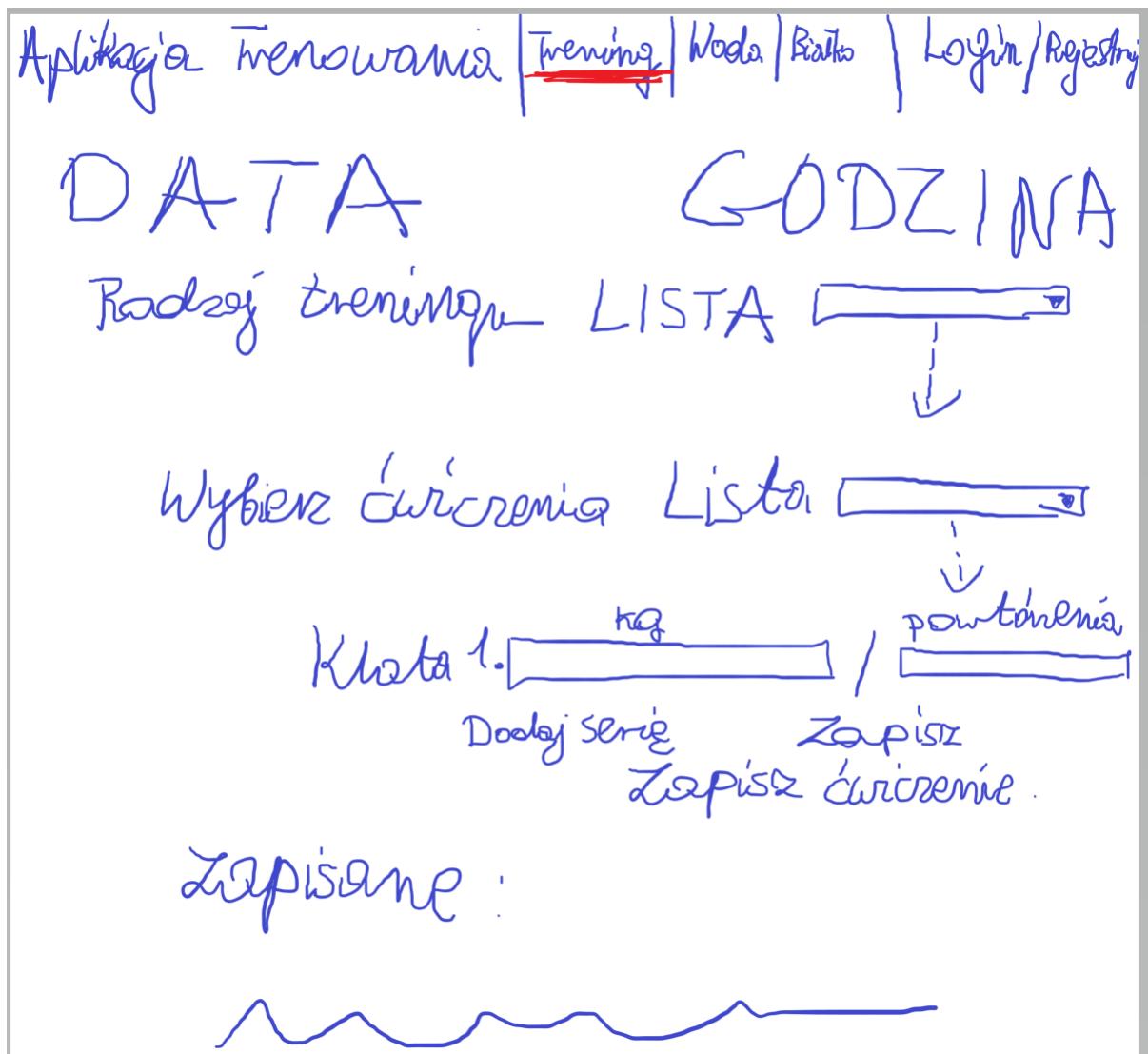
Select all Type ?

Find a collaborator...

DawidBilUP Collaborator

< Previous Next >

Przygotowaliśmy początkowe szkice które miały za zadanie rozplanować działanie aplikacji



Aplikacja treningowa | trening | Nada | Bialko } Login/Rejestracja

DATA
Dodałeś wypitą wołkę (ml)

zapisz

Wypita wołka

----- ml

Dzienny cel wołki

- ----- ml

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

DATA

Dodaj Zjedzone białko (g)



zapisz

Zjedzone białko

----- g

Dzienny cel białka

----- g

HomeController.cs* □ X

AplikacjaTrenowania

```
{\} 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();
            }
        }
    }
```

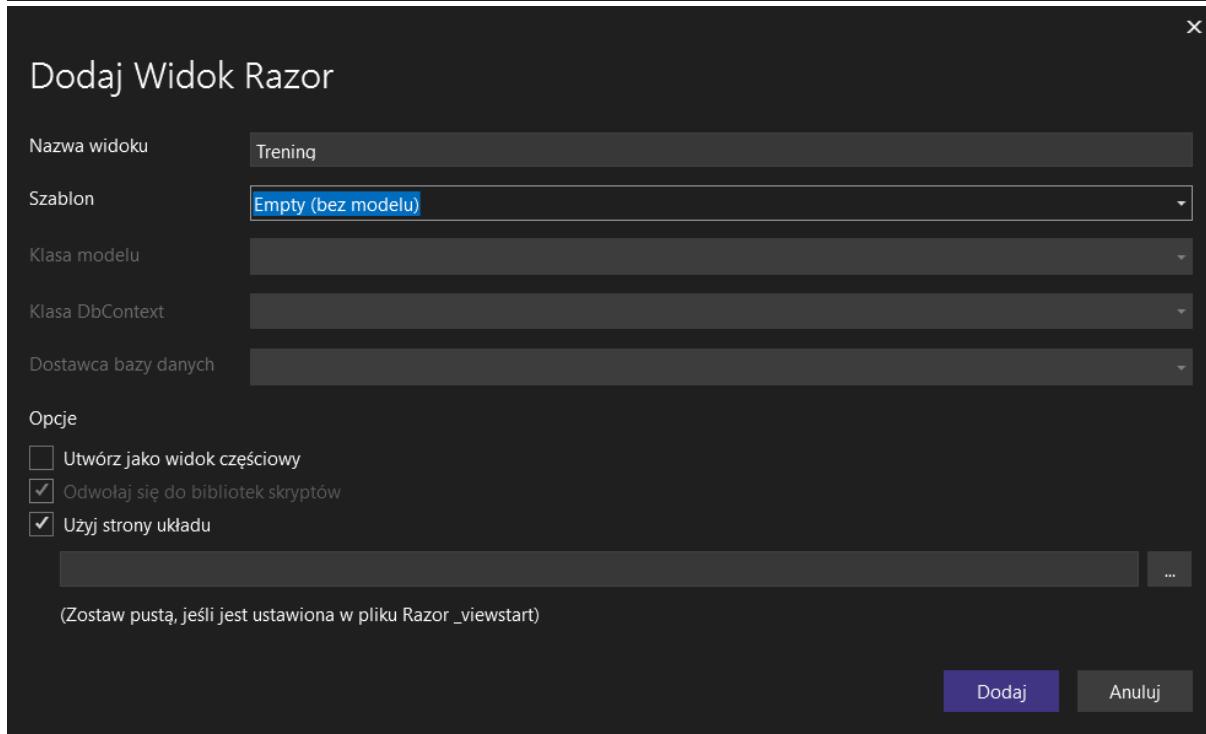
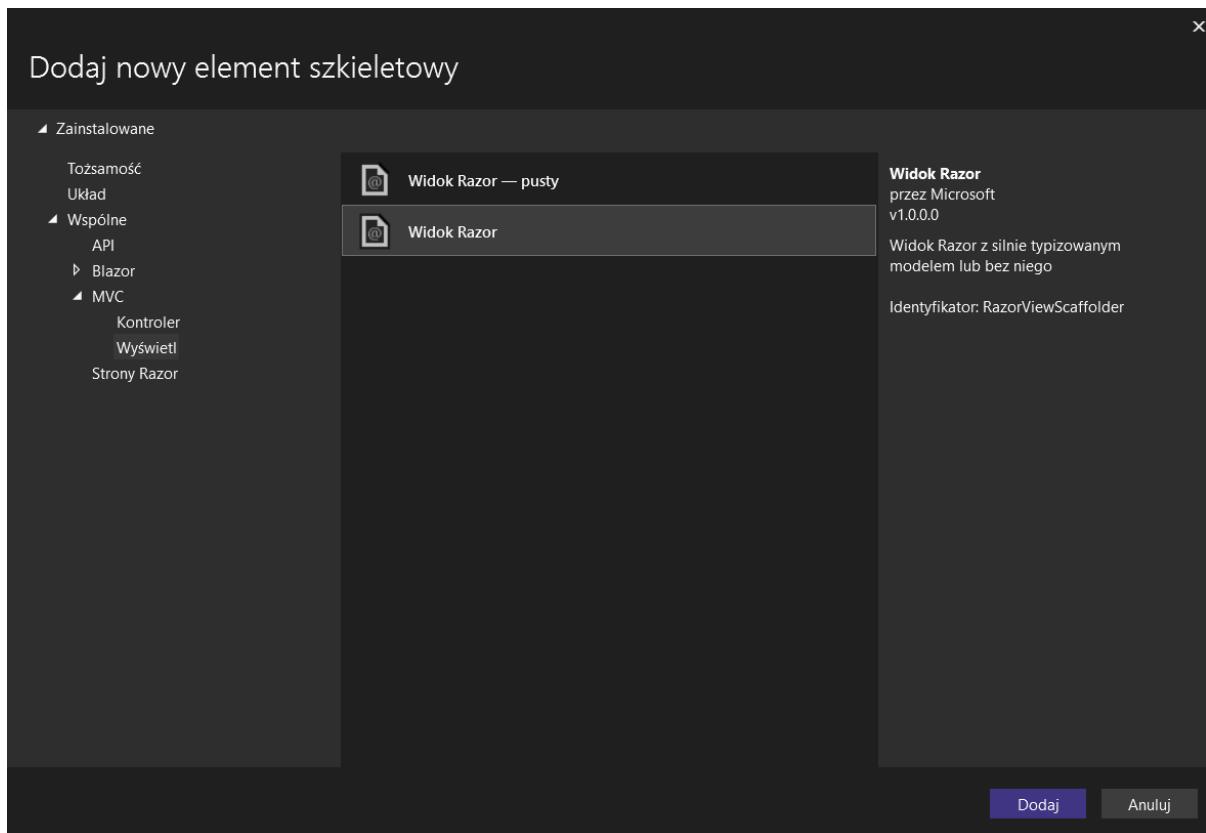
Odwołania: 0

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

Odwołania: 0

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

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



The screenshot shows the Visual Studio interface with two tabs open: "Layout.cshtml" and "Trening.cshtml". The "Layout.cshtml" tab displays the HTML structure of the application's layout, including the head section with meta tags, viewport settings, and links to Bootstrap CSS and the application's styles. It also shows the main body structure with a header containing a navigation bar. The "Trening.cshtml" tab is visible in the background. Below the tabs is a browser-like preview window showing the rendered navigation bar with links to "Home", "Privacy", and "Trening".

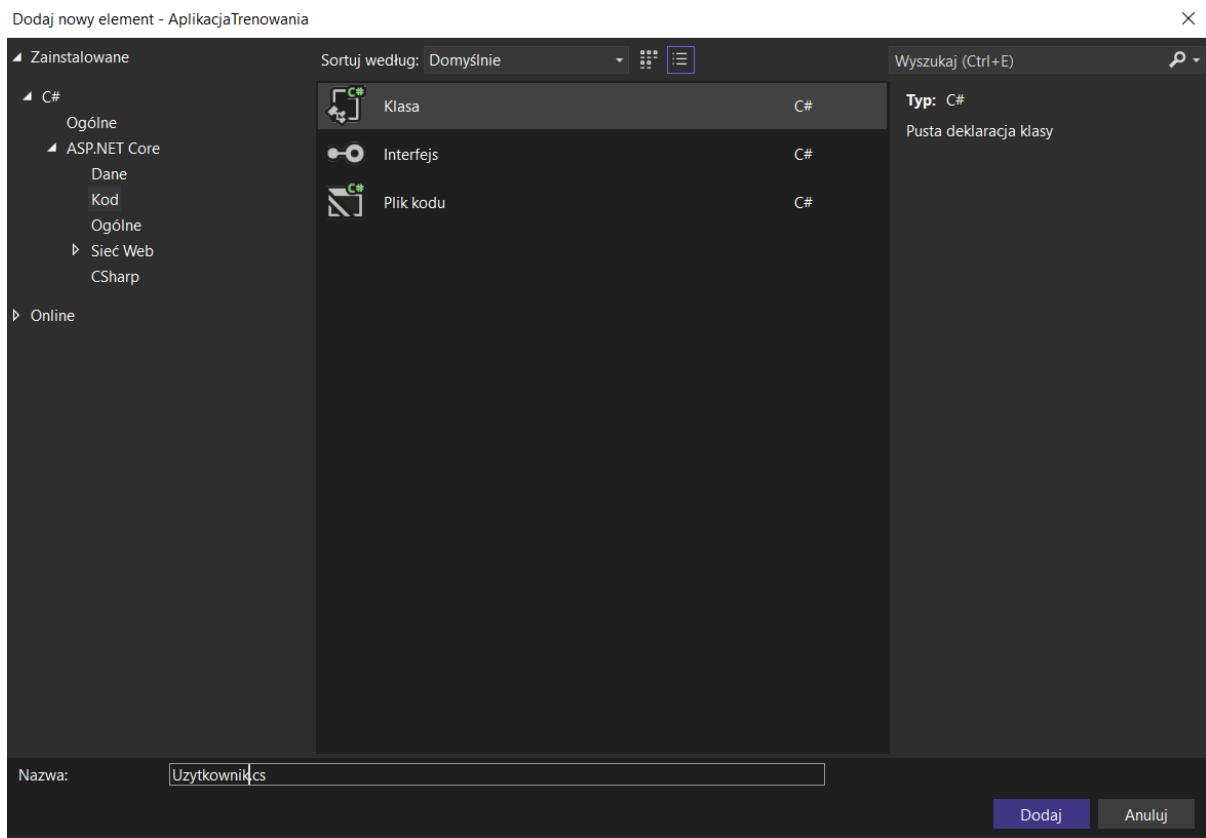
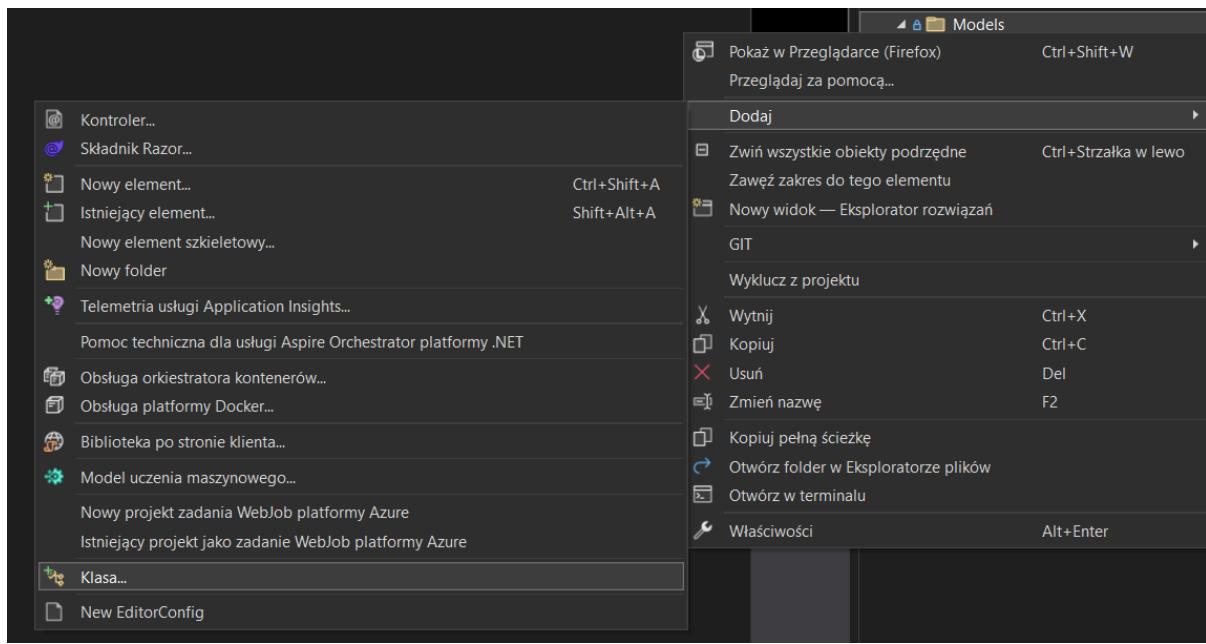
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ ViewData["Title"] - AplikacjaTrenowania</title>
    <link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />
    <link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />
    <link rel="stylesheet" href="/AplikacjaTrenowania.styles.css" asp-append-version="true" />
</head>
<body>
    <header>
        <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-white border-bottom box-shadow mb-3">
            <div class="container-fluid">
                <a class="navbar-brand" asp-area="" asp-controller="Home" asp-action="Index">AplikacjaTrenowania</a>
                <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target=".navbar-collapse" aria-expanded="false" aria-label="Toggle navigation">
                    <span class="navbar-toggler-icon"></span>
                </button>
                <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
                    <ul class="navbar-nav flex-grow-1">
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Index">Home</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Privacy">Privacy</a>
                        </li>
                        <li class="nav-item">
                            <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-action="Trening">Trening</a>
                        </li>
                    </ul>
                </div>
            </div>
        </nav>
    </header>
    <div class="container">
        <main>
            <h1>Trening</h1>
        </main>
    </div>
</body>
```

AplikacjaTrenowania Home Privacy Trening

Trening

Analogicznie dodaliśmy podstrony Woda i Białko

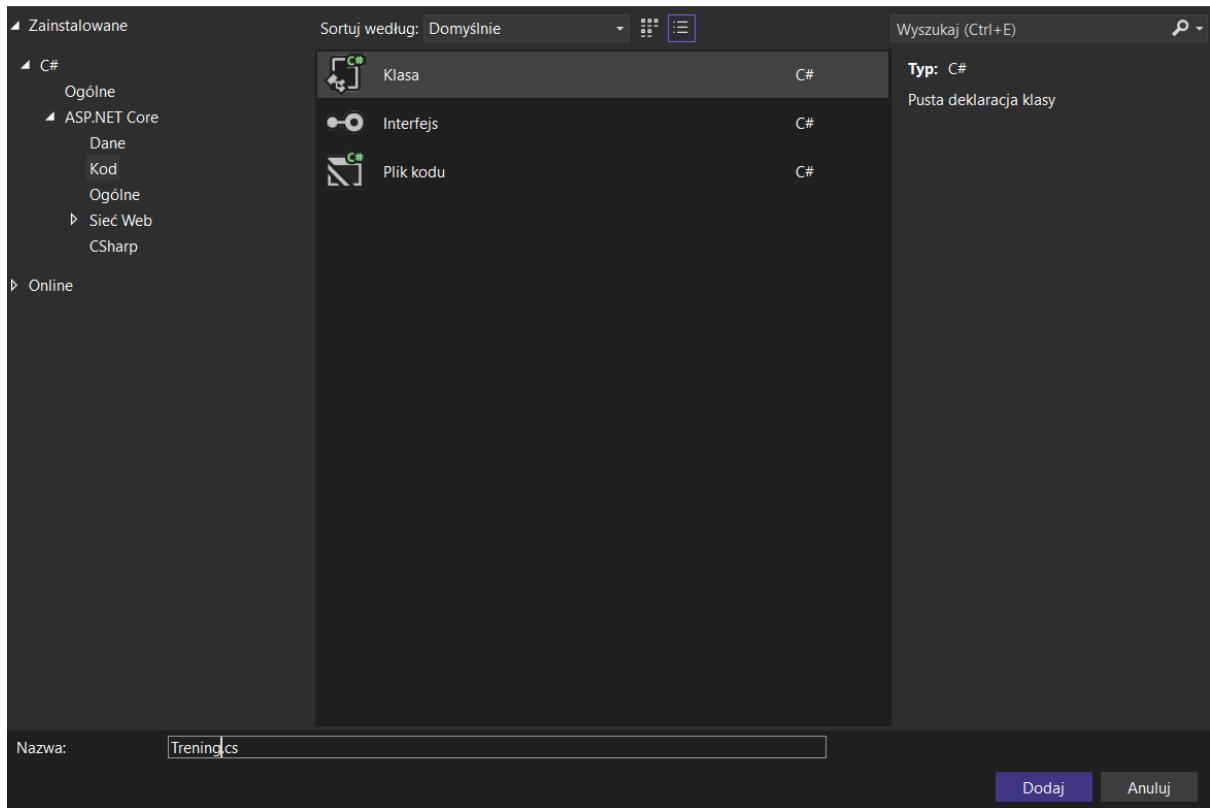
AplikacjaTrenowania Home Privacy Trening Woda Białko



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

23.12.2024

Dodaj nowy element - AplikacjaTrenowania



Trening.cs* X

AplikacjaTrenowania

ApplikacjaTrenowania.Models.Trening

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

namespace AplicacjaTrenowania.Models
{
    public class Trening
    {
        [Key]
        public int IdTreningu { get; set; }
        [Column(TypeName = "nvarchar(50)")]
        public string Data { get; set; }
        [Column(TypeName = "nvarchar(50)")]
        public string Godzina { get; set; }
        [Column(TypeName = "nvarchar(100)")]
        public string RodzajTreningu { get; set; }
        [Column(TypeName = "nvarchar(100)")]
        public string WybierzCwiczenie { get; set; }
        public List<Seria> Serie { get; set; } = new List<Seria>();
    }

    public class Seria
    {
        [Key]
        public int IdSerii { get; set; }
        [Column(TypeName = "int")]
        public int Kg { get; set; }
        [Column(TypeName = "int")]
        public int Powtorzenia { get; set; }
    }
}
```

The screenshot shows the Microsoft Visual Studio IDE interface. The main window displays the code for a file named `Trening.cs`. The code defines two classes: `Trening` and `Seria`, both annotated with `[Key]` from the `System.ComponentModel.DataAnnotations` namespace. The `Trening` class has properties `Data`, `Godzina`, `RodzajTreningu`, and `WybierzCwiczenie`. The `Seria` class has properties `IdSerii`, `Kg`, and `Powtorzenia`. The code editor highlights the `WybierzCwiczenie` property with a red squiggle underlining, indicating a potential issue. A tooltip provides a detailed error message:

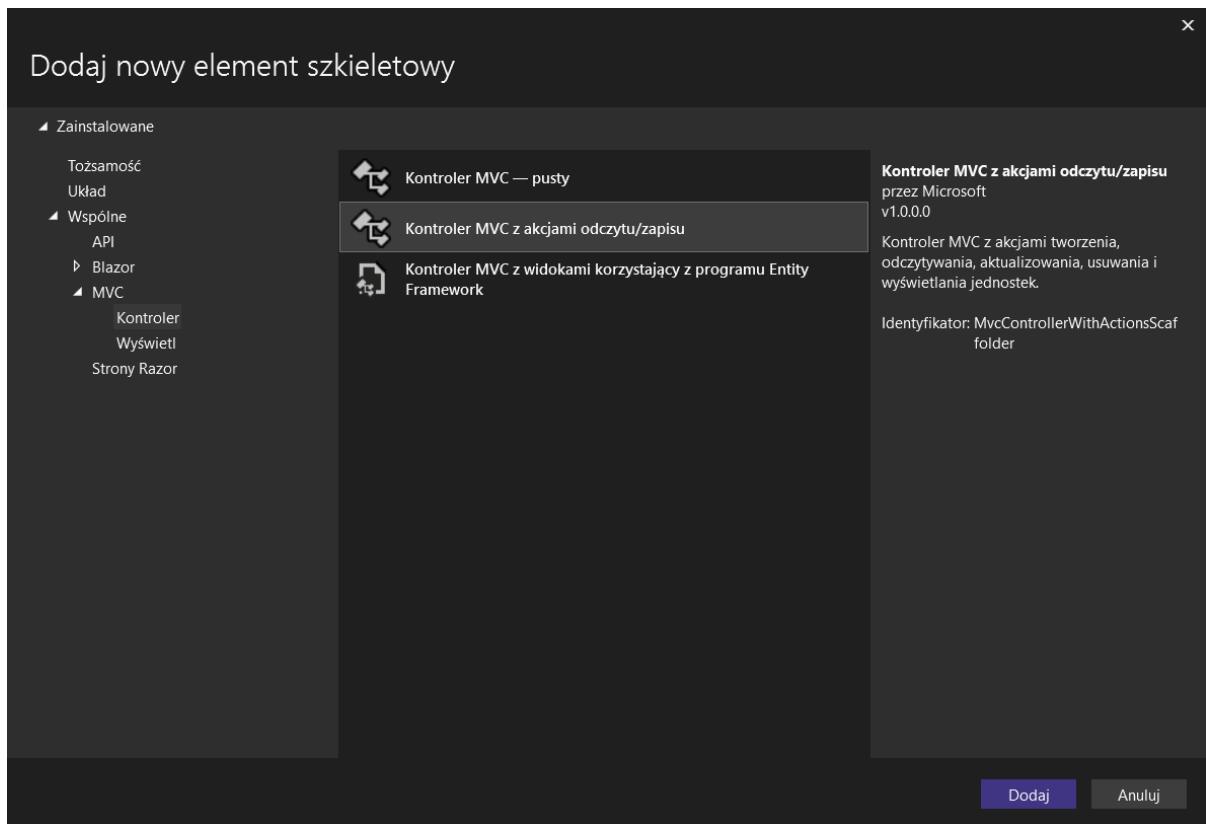
CS8618: Niedopuszczający wartości null element właściwości „WybierzCwiczenie” musi zawierać wartość inną niż null podczas działania konstruktora. Rozważ dodanie modyfikatora „required” lub zadeklarowanie właściwość jako dopuszczającego wartość null.

Pokaz potencjalne rozwiązania (Alt+Enter lub Ctrl+.)

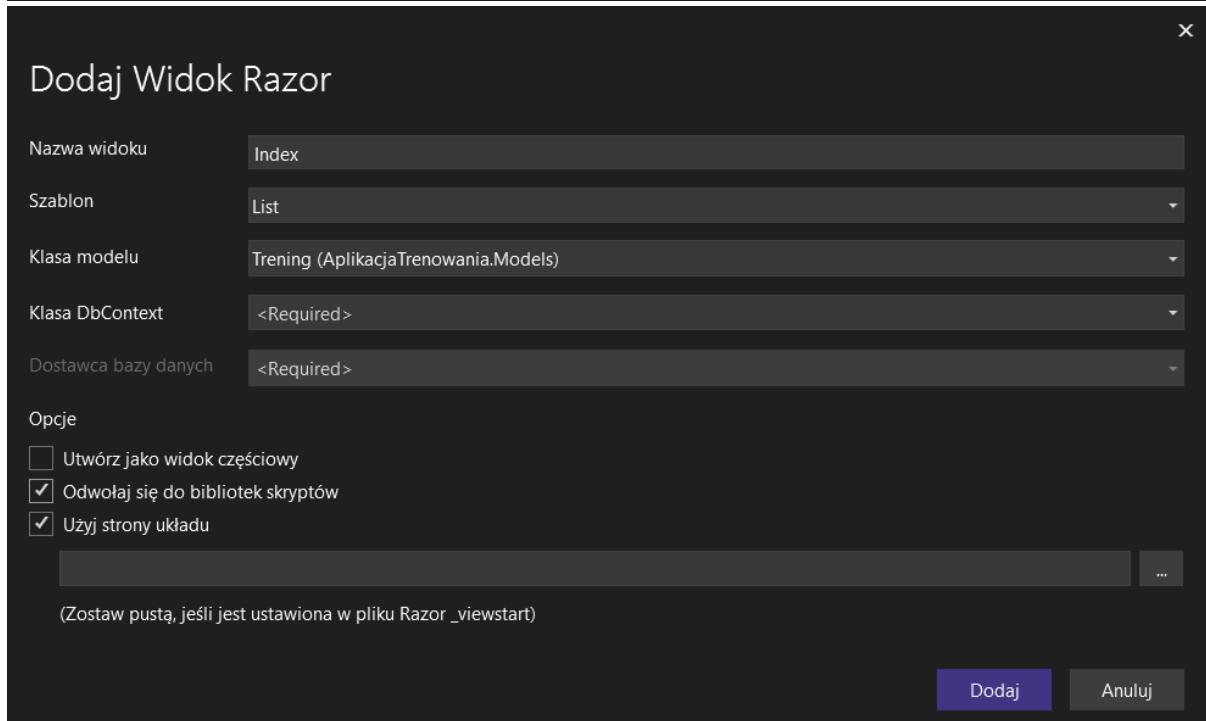
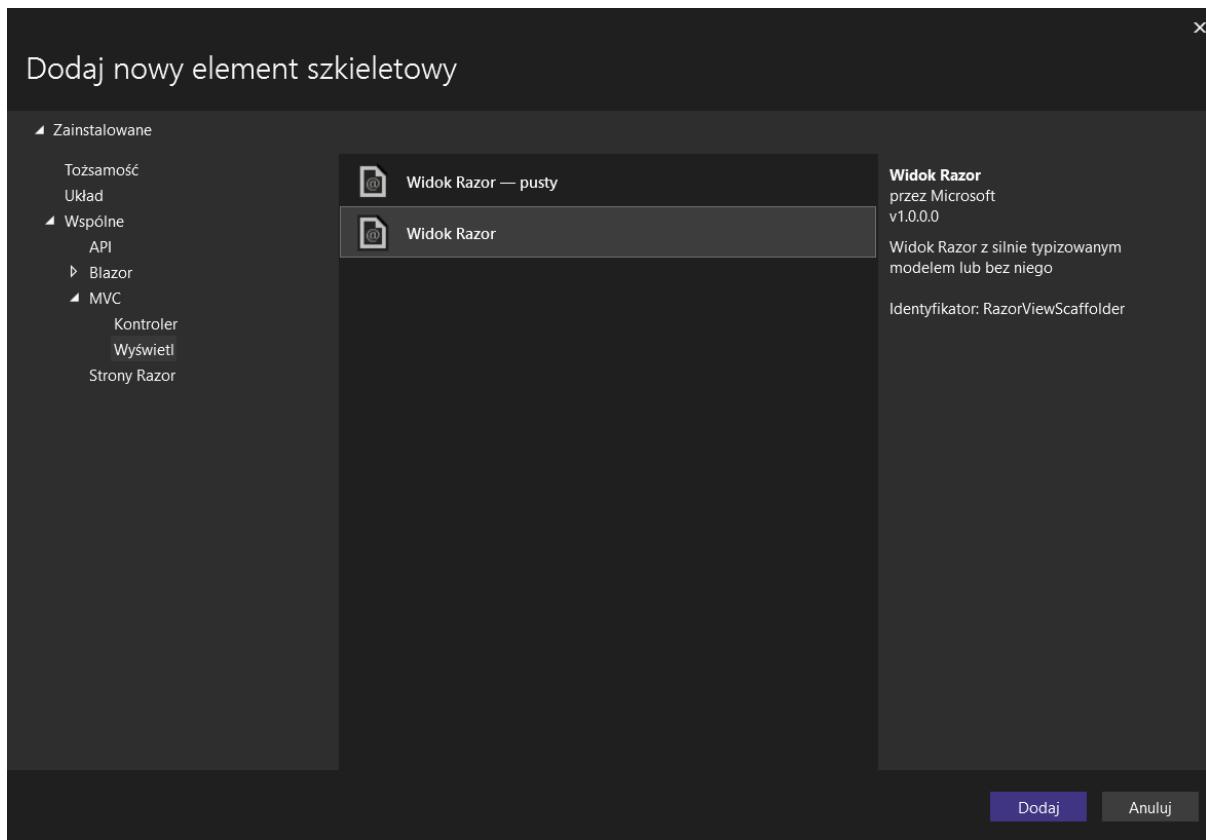
The status bar at the bottom of the IDE shows the text "Analiza" (Analysis).

```
Trening.cs* X
ApplikacjaTrenowania          ↳ 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; }
    }
  }
}
```



Usuwanie starego widoku trening



Pakiety pośrednie (1)

Microsoft.EntityFrameworkCore.Design przez Microsoft

Shared design-time components for Entity Framework Core tools.

Podgląd zmian

Program Visual Studio wprowadzi zmiany w tym rozwiązańiu.

AplikacjaTrenowania

Instalowanie: Microsoft.EntityFrameworkCore.Design.8.0.11

Nie pokazuj tego więcej Zastosuj Anuluj

Licencja na każdy pakiet jest udzielana przez jego właściciela. Firma NuGet nie ponosi odpowiedzialności za pakietów innych firm ani nie udziela na nie żadnych licencji.

Nie pokazuj tego więcej

Microsoft.EntityFrameworkCore.Design

nuget.org

Zainstalowane: 8.0.11

Wersja: 8.0.11

Mapowanie źródła pakietu jest wyłączone. Konfiguruj

Opcje

Opis

Shared design-time components for Entity Framework Core tools.

Wersja: 8.0.11

Autorzy: Microsoft

Licencja: MIT

Plik readme: Wyświetl plik readme

Data publikacji: wtorek, 12 listopada 2024 (12.11.2024)

Adres URL projektu: <https://docs.microsoft.com/en/core/>

Zgłos nadzorcze: <https://www.nuget.org/packages/Microsoft.EntityFrameworkCore.Design/8.0.11/ReportAbuse>

Tagi: Entity, Framework, Core, entity-framework-core, EF, Data, O/RM, EntityFramework, EntityFrameworkCore, EFCore

Zależności

- net5.0
Microsoft.EntityFrameworkCore.Relational (>= 8.0.11)
Microsoft.EntityFrameworkCore >= 2.14.3
Microsoft.CodeAnalysis.CSharp.Workspaces (>= 4.5.0)
Microsoft.Extensions.DependencyInjectionModel (>= 8.0.2)
Mono.TextTemplating (>= 2.2.1)

Microsoft.EntityFrameworkCore.Tools

nuget.org

Zainstalowane: 8.0.11

Wersja: 8.0.11

Mapowanie źródła pakietu jest wyłączone. Konfiguruj

Opcje

Dodaj Widok Razor

Nazwa widoku: Index

Szablon: List

Klasa modelu: Trening (AplikacjaTrenowania.Models)

Klasa DbContext: ApplicationDbContext (AplikacjaTrenowania.Areas.Identity.Data)

Dostawca bazy danych: Skonfigurowane z wybranego obiektu DbContext

Opcje:

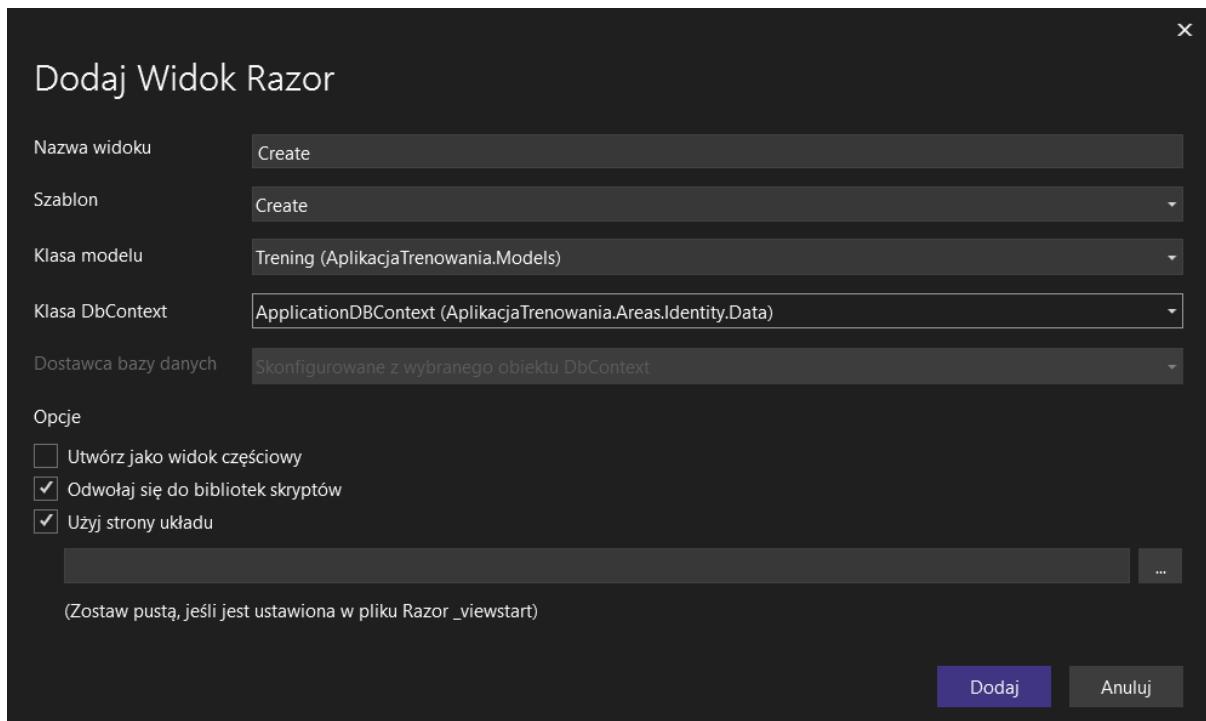
Utwórz jako widok częściowy

Odwołaj się do bibliotek skryptów

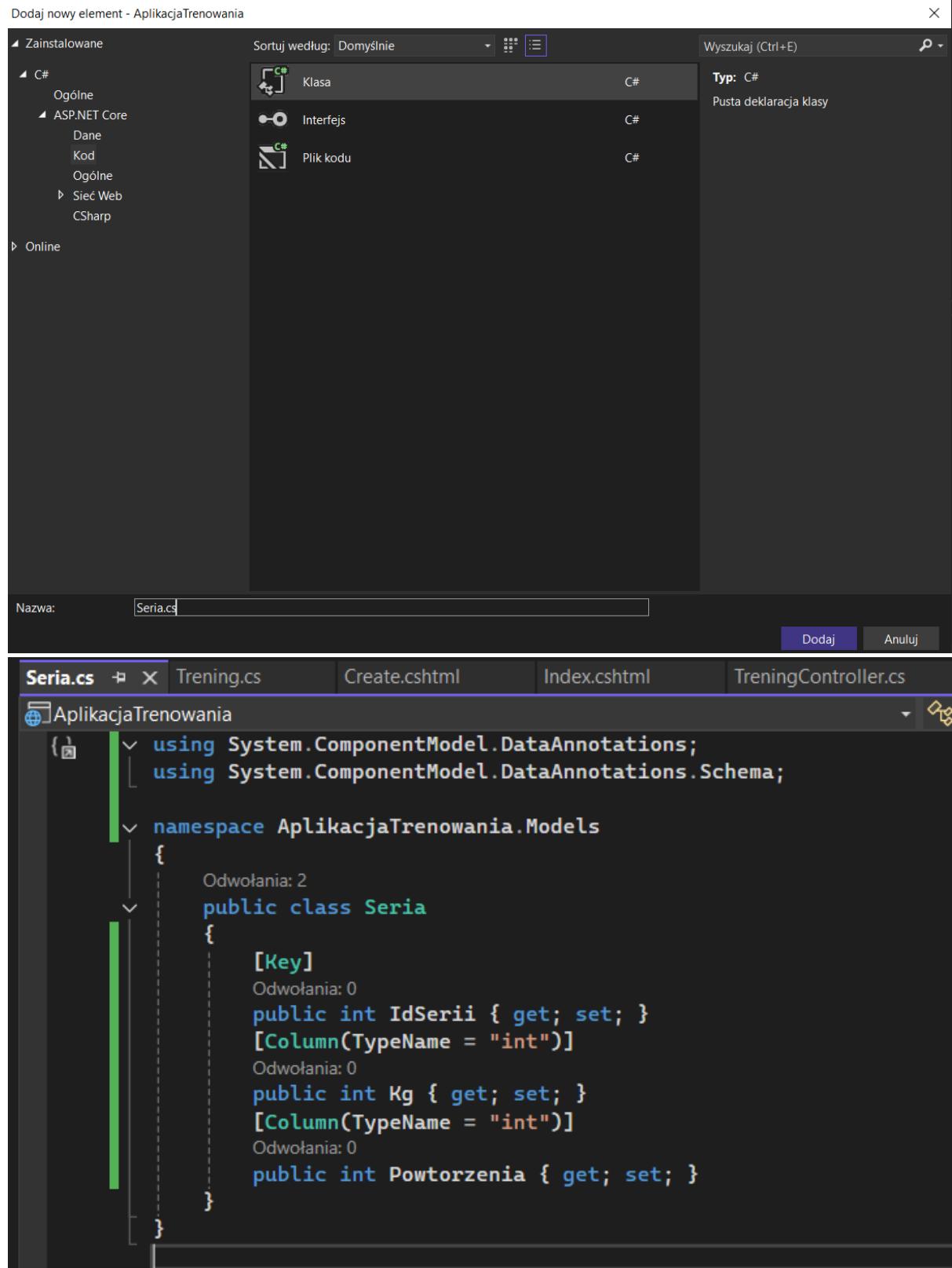
Użyj strony układu

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

Dodaj Anuluj



Popełniliśmy 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



```

Program.cs*  ✎ X Seria.cs  Trening.cs  Create.cshtml  Index.cshtml  TreningController.cs
ApplikacjaTrenowania
{
    using Microsoft.AspNetCore.Identity;
    using Microsoft.EntityFrameworkCore;
    using ApplikacjaTrenowania.Areas.Identity.Data;
    var builder = WebApplication.CreateBuilder(args);
    var connectionString = builder.Configuration.GetConnectionString("ApplicationDBCont

    builder.Services.AddDbContext<ApplicationDbContext>(options => options.UseSqlServer

    builder.Services.AddDefaultIdentity< ApplicationUser >(options => options.SignIn.Requ

    // Add services to the container.
    builder.Services.AddControllersWithViews();

    var app = builder.Build();

    // Configure the HTTP request pipeline.
    if (!app.Environment.IsDevelopment())
    {
        app.UseExceptionHandler("/Home/Error");
        // The default HSTS value is 30 days. You may want to change this for production
        app.UseHsts();
    }

    app.UseHttpsRedirection();
    app.UseStaticFiles();

    app.UseRouting();

    app.UseAuthorization();

    app.MapControllerRoute(
        name: "default",
        pattern: "{controller=Trening}/{action=Index}/{id?}");

    app.Run();
}

```

The screenshot shows the Visual Studio IDE with the code editor open to Program.cs. The code defines a WebApplication instance, sets up the database context, adds services for Identity and controllers, and configures the HTTP request pipeline. It includes middleware for exception handling, HSTS, HTTPS redirection, static files, routing, authorization, and a controller route for the 'Trening' controller.

Below the code editor, the browser window displays the application's index page. The page features a header with the title 'Index' and a table listing training data. A dropdown menu labeled 'Wybierz zawody' (Select competition) is visible, with options including 'AplikacjaTrenowania.dll', 'Otwórz istniejące wyjście', and 'Edytuj warunki'.

03.01.25

```
form.js          addserie.js        Program.cs        Index.cshtml      Create.cshtml      _Layout.cshtml     Trening.cs        AplikacjaTrenowania.Models.TreningController.cs
ApplikacjaTrenowania
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace AplikacjaTrenowania.Models
5  {
6      public class Trening
7      {
8          [Key]
9          public int IdTreningu { get; set; }
10         [Column(TypeName = "nvarchar(50)")]
11         public string Data { get; set; } = DateTime.Now.Date.ToString("d");
12         [Column(TypeName = "nvarchar(50)")]
13         public string Godzina { get; set; } = DateTime.Now.ToString("HH:mm");
14         [Column(TypeName = "nvarchar(100)")]
15         public string RodzajTreningu { get; set; } = string.Empty;
16         [Column(TypeName = "nvarchar(100)")]
17         public string WybierzCwiczenie { get; set; } = string.Empty;
18         public List<Seria> Serie { get; set; } = new List<Seria>();
19     }
20 }
21
```

```
formjs addserie.js Program.cs Index.cshtml Create.cshtml Layout.cshtml Trening.cs TreningController.cs
AplikacjaTrenowania
1 @using Microsoft.AspNetCore.Http;
2 using Microsoft.AspNetCore.Mvc;
3 using AplikacjaTrenowania.Models;
4
5 namespace AplikacjaTrenowania.Controllers
6 {
7     public class TreningController : Controller
8     {
9         private readonly ILogger<TreningController> _logger;
10        private static List<Trening> zapisaneTreningi = new List<Trening>();
11        private readonly Dictionary<string, List<string>> cwiczenia = new Dictionary<string, List<string>>
12        {
13            { "Siłowy", new List<string> { "Przysiady", "Martwy ciąg", "Wyciskanie sztangi" } },
14            { "Kardio", new List<string> { "Bieganie", "Rower", "Skakanka" } },
15            { "Stretching", new List<string> { "Joga", "Rozciąganie dynamiczne", "Pilates" } }
16        };
17        public TreningController(ILogger<TreningController> logger)
18        {
19            _logger = logger;
20        }
21        // GET: TreningController
22        1 odwołanie
23        public ActionResult Index()
24        {
25            return View();
26        }
27
28        // GET: TreningController/Details/5
29        Odwołania: 0
30        public ActionResult Details(int id)
31        {
32            return View();
33        }
34
35        // GET: TreningController/Create
36        Odwołania: 0
37        public ActionResult Create()
38        {
39            ViewBag.Cwiczenia = cwiczenia;
40            return View(new Trening());
41        }
42
43        // POST: TreningController/Create
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59 %  Nie znaleziono żadnych problemów |  Autopaste |  X
```

Screenshot of a code editor showing three tabs: TreningController.cs, Layout.cshtml*, and Layout.cshtml.

TreningController.cs

```

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 Odwołania: 0
55 public IActionResult ZapiszTrening(Trening model, int[] serieKg, int[] seriePowtorzenia)
56 {
57     for (int i = 0; i < serieKg.Length; i++)
58     {
59         model.Serie.Add(new Seria
60         {
61             Kg = serieKg[i],
62             Powtorka = seriePowtorzenia[i]
63         });
64     }
65     zapisaneTreningi.Add(model);
66     return RedirectToAction("ListaTreninguow");
67 }
68 }
69 
```

Layout.cshtml*

```

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 
```

Layout.cshtml

```

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" aria-expanded="false" aria-label="Toggle navigation">
17                     <span class="navbar-toggler-icon"></span>
18                 </button>
19                 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20                     <ul class="navbar-nav flex-grow-1">
21                         <li class="nav-item">
22                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Index">Trening</a>
23                         </li>
24                         <li class="nav-item">
25                             <a class="nav-link text-dark" asp-area="" asp-controller="Woda" asp-action="Index">Woda</a>
26                         </li>
27                         <li class="nav-item">
28                             <a class="nav-link text-dark" asp-area="" asp-controller="Bialko" asp-action="Index">Bialko</a>
29                         </li>
30                     </ul>
31                 </div>
32             </div>
33         </nav>
34     </header>
35     <div class="container">
36         <main role="main" class="pb-3">
37             @RenderBody()
38         </main>
39     </div>
40 
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     </footer>
46 
```

Bottom status bar: Nie znaleziono żadnych problemów

```
form.js | addserie.js | Program.cs | Index.cshtml | Create.cshtml ✘ X | _Layout.cshtml* | Trening.cs | TreningController.cs
AplikacjaTrenowania
1 @model AplikacjaTrenowania.Models.Trening
2
3 @{
4     ViewData["Title"] = "Create";
5 }
6
7 <h1>Create</h1>
8
9 <h4>Trening</h4>
10 <br />
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="Powyższenia" 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             </div>
61             <button type="button" id="addSeria" class="btn btn-primary mt-3">Dodaj serię</button>
62             <button type="submit" class="btn btn-success mt-3">Zapisz Trening</button>
63         </form>
64     </div>
65 </div>
66 <div>
67     <a asp-action="Index">Back to List</a>
68 </div>
69
70 <script src="~/js/form.js"></script>
71 <script src="~/js/addserie.js"></script>
72
73 @section Scripts {
74     @await Html.RenderPartialAsync("_ValidationScriptsPartial");
75 }
76
77
```

ApplikacjaTrenowania

```

1     @model IEnumerable<ApplikacjaTrenowania.Models.Trening>
2
3     @{
4         ViewData["Title"] = "Index";
5     }
6
7     <h1>Index</h1>
8
9     <p>
10    <a href="#">Create New</a>
11
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                <th></th>
28            </tr>
29        </thead>
30        <tbody>
31        @if (Model != null)
32        {
33            @foreach (var item in Model)
34            {
35                <tr>
36                    <td>
37                        @Html.DisplayFor(modelItem => item.Data)
38                    </td>
39                    <td>
40                        @Html.DisplayFor(modelItem => item.Godzina)
41                    </td>
42                    <td>
43                        @Html.DisplayFor(modelItem => item.RodzajTreningu)
44                    </td>
45                    <td>
46                        @Html.DisplayFor(modelItem => item.WybierzCwiczenie)
47                    </td>
48                    <td>
49                        <a href="#">Edit</a> |
50                        <a href="#">Details</a> |
51                        <a href="#">Delete</a>
52                    </td>
53                }
54            </tr>
55        }
56    </tbody>
57 </table>

```

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

ApplikacjaTrenowania

```

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

```

ApplikacjaTrenowania

```

1  using Microsoft.AspNetCore.Identity;
2  using Microsoft.EntityFrameworkCore;
3  using ApplikacjaTrenowania.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();

```

```
form.js addserie.js Program.cs Index.cshtml Create.cshtml _Layout.cshtml* Trening.cs TreningController.cs
ApplikacjaTrenowania Pliki zawartości języka JavaScript <global>
1  document.getElementById('addSerie').addEventListener('click', function () {
2      console.log('Dodał serię');
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="serię">
9              <input type="number" placeholder="Kg" class="form-control seria-kg" />
10             </div>
11             <div class="serię">
12                 <input type="number" placeholder="Powtóżenia" class="form-control seria-powtóżenia" />
13             </div>
14             <div class="serię">
15                 <button type="button" class="btn btn-danger remove-seria">Usuń</button>
16             </div>
17         `;
18         serieContainer.appendChild(seria);
19
20         // Dodaj obsługę usuwania serii
21         seria.querySelector('.remove-seria').addEventListener('click', function () {
22             seria.remove();
23         });
24     });
}
form.js Program.cs Index.cshtml Create.cshtml _Layout.cshtml* Trening.cs TreningController.cs
ApplikacjaTrenowania 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](#)

20250105111703_Initial.cs Save.cshtml appsettings.json ApplicationDbContext.cs Trening.cs Seria.cs Program.cs _Layout.cshtml

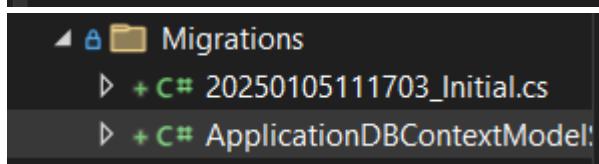
```

AplikacjaTrenowania
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using AplikacjaTrenowania.Models;
4
5  namespace AplikacjaTrenowania.Areas.Identity.Data;
6
7  public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
8  {
9      public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
10     : base(options)
11     {
12     }
13
14     public DbSet<Trening> Trening { get; set; }
15     public DbSet<Seria> Seria { get; set; }
16
17 }
```

addserie.js form.js Create.cshtml TreningController.cs Save.cshtml appsettings.json ApplicationDbContext.cs Trening.cs Seria.cs Program.cs _Layout.cshtml

```

AplikacjaTrenowania
1  namespace AplikacjaTrenowania.Controllers;
2
3  public class TreningController : Controller
4  {
5      private readonly ApplicationDbContext _context;
6
7      public TreningController(ApplicationDbContext context)
8      {
9          _context = context;
10     }
11
12     private readonly ILogger<TreningController> _logger;
13     private static List<Trening> zapisaneTreningi = new List<Trening>();
14     private readonly Dictionary<string, List<string>> cwiczenia = new() {
15         { "Siłowy", new List<string> { "Przysiady", "Marthy clap", "Wyciskanie sztangi" } },
16         { "Wardio", new List<string> { "Bieganie", "Rowing", "Skakanka" } },
17         { "Stretching", new List<string> { "Joga", "Rozciąganie dynamiczne", "Pilates" } }
18     };
19
20     // GET: Trening
21     public ActionResult Index() => View();
22
23     // GET: Trening/Details/5
24     public ActionResult Details(int id) => View();
25
26     // GET: Trening/Create
27     public ActionResult Create()
28     {
29         ViewBag.Cwiczenia = cwiczenia;
30         return View(new Trening());
31     }
32
33     [HttpPost]
34     public ActionResult Create([Bind(Include = "Nazwa, Czas, Opis, Cwiczenia, IdSerii, IdTreningu, Powtorzenia")]
35     Trening model) => RedirectToAction("Save", model);
36
37     // GET: Trening/Save
38     public ActionResult Save(Trening model) => View(model);
39
40 }
```



addserie.js form.js Create.cshtml TreningController.cs Save.cshtml appsettings.json ApplicationDbContext.cs Trening.cs Seria.cs Program.cs _Layout.cshtml

```

AplikacjaTrenowania
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace AplikacjaTrenowania.Models;
5
6  public class Seria
7  {
8      [Key]
9      public int IdSerii { get; set; }
10     [Column(TypeName = "int")]
11     public int Kg { get; set; }
12     [Column(TypeName = "int")]
13     public int Powtorzenia { get; set; }
14     [ForeignKey("Trening")]
15     public int IdTreningu { get; set; }
16     public Trening Trening { get; set; }
17 }
```

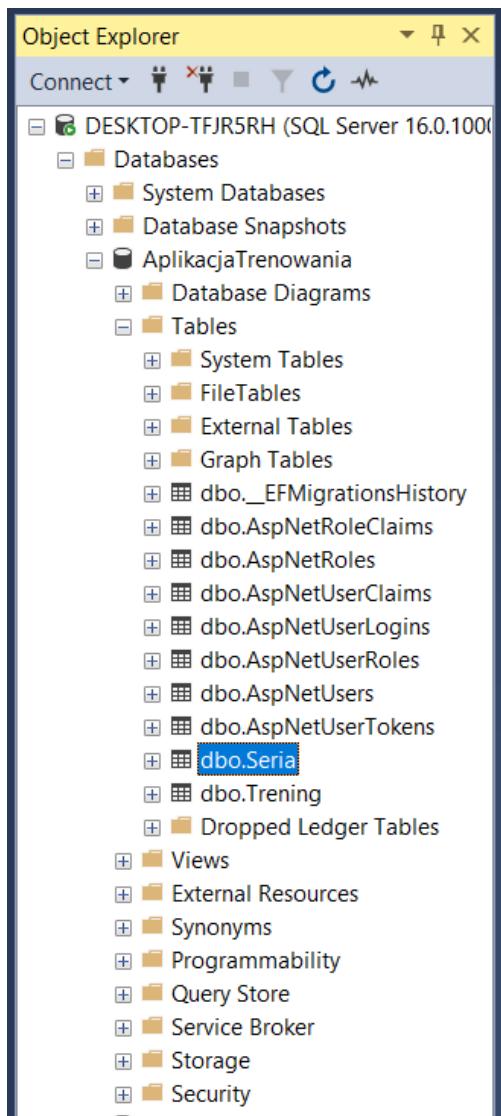
```
ApplicationDB...elSnapshot.cs 20250105111703_Initial.cs Save.cshtml appsettings.json ApplicationDbContext.cs Trening.cs Seria.cs Program.cs _Layout.cshtml
AplikacjaTrenowania
1  using System.ComponentModel.DataAnnotations;
2  using System.ComponentModel.DataAnnotations.Schema;
3
4  namespace AplikacjaTrenowania.Models
5  {
6      Odwołania: 17
7      public class Trening
8      {
9          [Key]
10         Odwołania: 4
11         public int IdTreningu { get; set; }
12         [Column(TypeName = "varchar(50)")]
13         Odwołania: 6
14         public string Data { get; set; } = DateTime.Now.Date.ToString("d");
15         [Column(TypeName = "varchar(50)")]
16         Odwołania: 6
17         public string Godzina { get; set; } = DateTime.Now.ToString("HH:mm");
18         [Column(TypeName = "varchar(100)")]
19         Odwołania: 5
20         public string RodzajTreningu { get; set; } = string.Empty;
21         [Column(TypeName = "nvarchar(100)")]
22         Odwołania: 4
23         public List<Seria> Serie { get; set; } = [];
24             1 odwołanie
25         public Trening() => Serie.Add(new Seria());
26     }
27 }
```

```
addserie.js form.js Create.cshtml TreningController.cs
ApplicationDB...elSnapshot.cs 20250105111703_Initial.cs Save.cshtml appsettings.json Application
AplikacjaTrenowania
10 <div>
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" hidden=@cwiczenie></option>
44                             }
45                         </select>
46                     </div>
47                     <div id="form-group-seria">
48                         <div class="seria-row">
49                             <div class="seryjka">
50                                 <input asp-for="Serie[0].Kg" class="form-control" />
51                             </div>
52                             <div class="seryjka">
53                                 <input asp-for="Serie[0].Powtorzenia" class="form-control" />
54                             </div>
55                             <div class="seryjka">
56                                 <button type="button" class="btn btn-danger remove-seria">Usuń</button>
57                             </div>
58                         </div>
59                     </div>
60                     <button type="button" id="addSerie" class="btn btn-primary mt-3">Dodaj serię</button>
61                     <button type="submit" class="btn btn-success mt-3">Zapisz Trening</button>
62                 </form>
63             </div>
64         </div>
65     </div>
66     <div>
67         <a asp-action="Index">Back to List</a>
68     </div>
69
70     <script src="/js/form.js"></script>
71     <script src="/js/addserie.js"></script>
72
73     @section Scripts {
74         @{
75             await Html.RenderPartialAsync("_ValidationScriptsPartial");
76         }
77     }
78 }
```

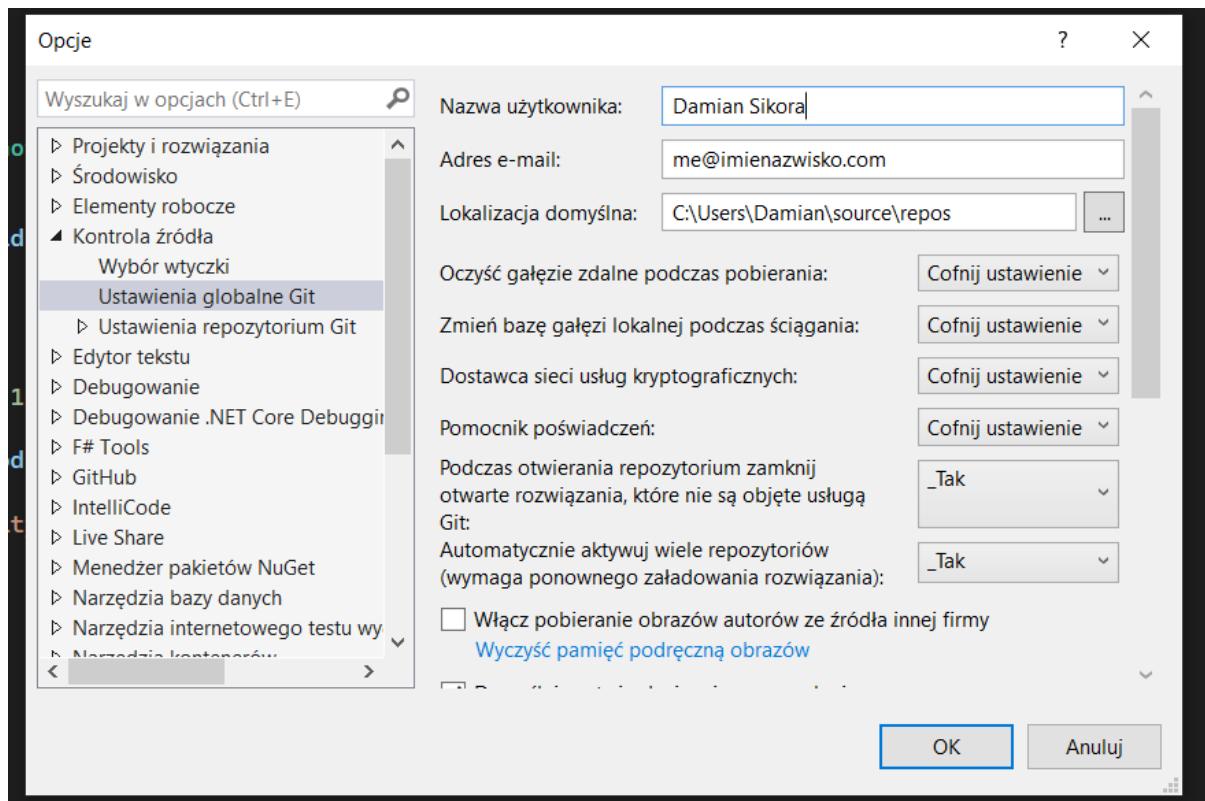
```
addserie.js  form.js  Create.cshtml  TreningController.cs  Save.cshtml  appsettings.json  ApplicationDbContext.cs  Trening.cs  Seria.cs  Program.cs  _Layout.cshtml
ApplicationDB...elSnapshot.cs  20250105111703_Initial.cs
ApplikacjaTrenowania
1  @model ApplikacjaTrenowania.Models.Trening
2
3  @{
4      ViewData["Title"] = "Save";
5  }
6
7  <h1>Save</h1>
8
9  <h4>Dodano trening @Model.IdTreningu @Model.Data @Model.Godzina @Model.RodzajTreningu @Model.WybierzCwiczenie. </h4>
10
11  <ul>
12      @foreach (var seria in Model.Serie)
13      {
14          <li>@seria.Kg @seria.Powtorzenia</li>
15      }
16  </ul>
17
18  <div>
19      <a href="#" asp-action="Index">Wyświetl listę</a>
20  </div>
21  <div>
22      <a href="#" asp-action="Create">Dodaj kolejny trening</a>
23  </div>
24
25  <script src="~/js/form.js"></script>
26  <script src="~/js/addserie.js"></script>
27
28  @section Scripts {
29      @{
30  }
31 }
```

```
addserie.js  form.js  Create.cshtml  TreningController.cs  Save.cshtml  appsettings.json  ApplicationDbContext.cs  Trening.cs  Seria.cs  Program.cs  _Layout.cshtml
ApplicationDB...elSnapshot.cs  20250105111703_Initial.cs
ApplikacjaTrenowania Plik zawartości język JavaScript
1  document.getElementById("addSerie").addEventListener("click", function () {
2      const container = document.getElementById("form-group-seria");
3      const rows = container.getElementsByClassName("seria-row");
4      const index = rows.length;
5      console.log(index);
6
7      const seria = document.createElement("div");
8      seria.className = "row seria-row";
9      seria.innerHTML =
10         <div class="serijka">
11             <input type="number" name="Serie[$index].Kg" placeholder="Kg" class="form-control seria-kg" />
12         </div>
13         <div class="serijka">
14             <input type="number" name="Serie[$index].Powtorzenia" placeholder="Powtórzenia" class="form-control seria-powtorzenia" />
15         </div>
16         <div class="serijka">
17             <button type="button" class="btn btn-danger remove-seria">Usuń</button>
18         </div>
19     ;
20     container.appendChild(seria);
21
22     seria.querySelector(".remove-seria").addEventListener("click", function () {
23         seria.remove();
24     });
25 });
```

```
addserie.js  form.js  Create.cshtml  TreningController.cs  Save.cshtml  appsettings.json  ApplicationDbContext.cs  Trening.cs  Seria.cs  Program.cs  _Layout.cshtml
ApplicationDB...elSnapshot.cs  20250105111703_Initial.cs
Schema: https://json.schemastore.org/appsettings.json
1  {
2      "Logging": {
3          "LogLevel": {
4              "Default": "Information",
5              "Microsoft.AspNetCore": "Warning"
6          }
7      },
8      "AllowedHosts": "*",
9      "ConnectionStrings": {
10          "ApplicationDbContextConnection": "Server=DESKTOP-TFJR5RH;Database=AplikacjaTrenowania;Trusted_Connection=True;TrustServerCertificate=True;"
11      }
12  }
```



5.01.2025



```
4     using Microsoft.EntityFrameworkCore;
5 
6     namespace AkademiaTrenowania.Controllers
7     {
8         Odwołanie: 2
9         public class TreningController : Controller
10        {
11            private readonly ApplicationDbContext _context;
12            Odwołanie: 0
13            public TreningController(ApplicationDbContext context)
14            {
15                _context = context;
16            }
17            private readonly ILogger<TreningController> _logger;
18            private static List<Trening> zapisaneTreningi = new List<Trening>();
19            private readonly Dictionary<string, List<string>> Ćwiczenia = new()
20            {
21                { "Silowy", new List<string> { "Przysiady", "Marwy ciąg", "Wyciskanie sztangi" } },
22                { "Kardio", new List<string> { "Bieganie", "Rowery", "Skakanka" } },
23                { "Stretching", new List<string> { "Joga", "Rozciąganie dynamiczne", "Pilates" } }
24            };
25            // GET: Trening
26            Odwołanie: 0
27            public ActionResult Index()
28            {
29                var trening = _context.Trening.Include(x => x.Serie);
30                var daty = trening.Select(x => x.Data).Distinct().ToList();
31                ViewBag.Daty = daty;
32                return View(trening);
33            }
34            // GET: Trening/Details/5
35            Odwołanie: 0
36            public ActionResult Details(int id) => View();
37            Odwołanie: 0
38            public ActionResult Delete(int id)
39            {
40                var trening = _context.Trening.Find(id);
41                if(trening != null) _context.Trening.Remove(trening);
42                _context.SaveChanges();
43                return RedirectToAction("Index");
44            }
45            // GET: Trening/Create
46            Odwołanie: 0
47            public ActionResult Create()
48            {
49                ViewBag.Ćwiczenia = Ćwiczenia;
50                return View(new Trening());
51            }
52            // POST: Trening/Create
53            [HttpPost]
54            Odwołanie: 0
55            public ActionResult Create(Trening model) {
56                _context.Trening.Add(model);
57                _context.SaveChanges();
58                return RedirectToAction("Save", new{id=model.IdTreningu});
59            }
60            // GET: Trening/Save
61            Odwołanie: 0
62            public ActionResult Save(int id)
63            {
64                var trening = _context.Trening.Include(x => x.Serie).FirstOrDefault(x => x.IdTreningu == id);
65                return View(trening);
66            }
67        }
68    }
69 }
```

```
addserie.js    form.js    Create.cshtml    TreningController.cs
ApplicationDB...elSnapshot.cs  20250105111703_Initial.cs  Save.cshtml  appsettings.json  Application
[App] AplikacjaTrenowania
1  @model AplikacjaTrenowania.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                 </label>
20             </div>
21             <div class="form-group">
22                 <label asp-for="Godzina" class="control-label"><label>
23                     <input asp-for="Godzina" class="form-control" readonly />
24                     <span asp-validation-for="Godzina" class="text-danger"></span>
25                 </label>
26             </div>
27             <div class="form-group">
28                 <label asp-for="RodzajTreningu" class="control-label"><label>
29                     <select id="rodzajTreningu" asp-for="RodzajTreningu" class="form-control">
30                         <option value="" disabled selected>Wybierz rodzaj treningu</option>
31                         @foreach (var rodzaj in ViewBag.Cwiczenia.Keys)
32                         {
33                             <option value="@{rodzaj}"@{rodzaj}></option>
34                         }
35                     </select>
36                 </div>
37                 <div class="form-group">
38                     <label asp-for="WybierzCwiczenie" class="control-label"><label>
39                     <select id="wybierzCwiczenie" asp-for="WybierzCwiczenie" class="form-control">
40                         <option value="" disabled selected>Wybierz Ćwiczenie</option>
41                         @foreach (var item in ViewBag.Cwiczenia)
42                         {
43                             <option data-cwiczenie="@item.Key" hidden=@cwiczenie></option>
44                         }
45                     </select>
46                 </div>
47                 <div id="form-group-seria">
48                     <button type="button" id="addSeria" class="btn btn-primary mt-3">Dodaj serię</button>
49                     <button type="submit" class="btn btn-success mt-3">Zapisz Trening</button>
50                 </div>
51             </form>
52         </div>
53     </div>
54 </div>
55
56     <div>
57         <a asp-action="Index">Back to List</a>
58     </div>
59
60     <script src="~/js/form.js"></script>
61     <script src="~/js/addserie.js"></script>
62
63     @section Scripts {
64         @{
65             await Html.RenderPartialAsync("_ValidationScriptsPartial");
66         }
67     }
68 }
```

The screenshot shows a code editor window with the file `Trening.cs` open. The code defines a class `Trening` with properties `IdTreningu`, `Data`, `Godzina`, `RodzajTreningu`, `WybierzCwiczenie`, and a list of `Seria`. The code uses `[Key]` and `[Column]` attributes from `System.ComponentModel.DataAnnotations` and `System.ComponentModel.DataAnnotations.Schema` namespaces.

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

```
Index.cshtml  ApplicationDB...elSnapshot.cs  20250105111703_Initial.cs
AplikacjaTrenowania
1  @model IEnumerable<AplikacjaTrenowania.Models.Trening>
2
3  @{
4      ViewData["Title"] = "Index";
5  }
6
7  <h1>Index</h1>
8
9  <p>
10     <a href="#" asp-action="Create">Create New</a>
11
12  <table class="table">
13      <thead>
14          <tr>
15              <th>
16                  @Html.DisplayNameFor(model => model.Data)
17              </th>
18          </tr>
19      </thead>
20      <tbody>
21          @if (Model != null)
22          {
23              @foreach(var data in ViewBag.Daty)
24              {
25                  <tr class="rozwijanie-przycisk-wiersz">
26                      <td>
27                          <button class="rozwijanie-przycisk">+</button>@data
28                      </td>
29                  </tr>
30                  <tr class="hidden">
31                      <th>
32                          @Html.DisplayNameFor(model => model.Godzina)
33                      </th>
34                      <th>
35                          @Html.DisplayNameFor(model => model.RodzajTreningu)
36                      </th>
37                      <th>
38                          @Html.DisplayNameFor(model => model.WybierzCwiczenie)
39                      </th>
40                  </tr>
41                  @foreach (var item in Model.Where(x => x.Data == data))
42                  {
43                      <tr class="hidden">
44                          <td>
45                              @Html.DisplayFor(modelItem => item.Godzina)
46                          </td>
47                          <td>
48                              @Html.DisplayFor(modelItem => item.RodzajTreningu)
49                          </td>
50                          <td>
51                              @Html.DisplayFor(modelItem => item.WybierzCwiczenie)
52                          </td>
53                          <td>
54                              <a href="#" asp-action="Edit" asp-route-id="@item.IdTreningu">Edit</a> |
55                              <a href="#" asp-action="Details" asp-route-id="@item.IdTreningu">Details</a> |
56                              <a href="#" asp-action="Delete" asp-route-id="@item.IdTreningu">Delete</a>
57                          </td>
58                      </tr>
59                  }
60              }
61          </tbody>
62      </table>
63  <section Scripts >
64      <script>
65          document.querySelectorAll(".rozwijanie-przycisk").forEach(button => {
66              button.addEventListener("click", function () {
67                  let nextRow = this.closest("tr").nextElementSibling;
68                  while (nextRow && !nextRow.classList.contains("rozwijanie-przycisk-wiersz")) {
69                      if (nextRow.classList.contains("hidden")) {
70                          nextRow.classList.remove("hidden");
71                          // nextRow.classList.add("visible");
72                      } else {
73                          // nextRow.classList.remove("visible");
74                          nextRow.classList.add("hidden");
75                      }
76                      nextRow = nextRow.nextElementSibling;
77                  }
78                  this.textContent = this.textContent === "+ ? -- : +";
79              });
80          });
81      </script>
82  </section>
```

```
Index.cshtml.css  Index.cshtml  ApplicationDB...elSnapshot.cs
1   .hidden{
2       display: none;
3   }
```

07.01.2025

```
addserie.js form.js Create.cshtml TreningController.cs expandingist.js deleteworkout.js
Repozytorium ...ania (master) Index.cshtml.cs Index.cshtml ApplicationDB...elSnapshot.cs 20250105111703_Initial.cs Save.cshtml appsettings.json ApplicationDBC
AplikacjaTrenowania
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  |</tr>
19  </thead>
20  <tbody>
21  <tr>
22  <td>
23  |   @foreach(var data in ViewBag.Daty)
24  |   {
25  |       <tr class="rozwijanie-przycisk-wiersz">
26  |           <td>
27  |               <button class="rozwijanie-przycisk">+</button>@data
28  |           </td>
29  |       </tr>
30  |       <tr class="hidden">
31  |           <th>
32  |               @Html.DisplayNameFor(model => model.Godzina)
33  |           </th>
34  |           <th>
35  |               @Html.DisplayNameFor(model => model.RodzajTreningu)
36  |           </th>
37  |           <th>
38  |               @Html.DisplayNameFor(model => model.WybierzCwiczenie)
39  |           </th>
40  |       </tr>
41  |       @foreach (var item in Model.Where(x => x.Data == data))
42  |       {
43  |           <tr class="hidden">
44  |               <td>
45  |                   @Html.DisplayFor(modelItem => item.Godzina)
46  |               </td>
47  |               <td>
48  |                   @Html.DisplayFor(modelItem => item.RodzajTreningu)
49  |               </td>
50  |               <td>
51  |                   @Html.DisplayFor(modelItem => item.WybierzCwiczenie)
52  |               </td>
53  |           <td>
54  |               <a asp-action="Edit" asp-route-id="@item.IdTreningu">Edit</a> |
55  |               <a asp-action="Details" asp-route-id="@item.IdTreningu">Details</a> |
56  |               <button onclick="deleteWorkout(@item.IdTreningu, this)">Delete</button>
57  |           </td>
58  |       </tr>
59  |     }
60  |   }
61  |</tbody>
62  </table>
63  <section Scripts >
64  <script src="/js/expandingist.js"></script>
65  <script src="/js/deleteworkout.js"></script>
66  </section>
67
68
```

```
addserie.js form.js Create.cshtml TreningController.cs expandingist.js deleteworkout.js
Repozytorium ...ania (master) Index.cshtml.cs Index.cshtml ApplicationDB...elSnapshot.cs 20250105111703_Initial.cs
AplikacjaTrenowania Pliki zawartości języka JavaScript <global>
1 document.querySelectorAll(".rozwijanie-przycisk").forEach(button => {
2   button.addEventListener("click", function () {
3     let nextRow = this.closest("tr").nextElementSibling;
4     while (nextRow && !nextRow.classList.contains("rozwijanie-przycisk-wiersz")) {
5       if (nextRow.classList.contains("hidden")) {
6         nextRow.classList.remove("hidden");
7         // nextRow.classList.add("visible");
8       } else {
9         // nextRow.classList.remove("visible");
10        nextRow.classList.add("hidden");
11      }
12      nextRow = nextRow.nextElementSibling;
13    }
14    this.textContent = this.textContent === "+" ? "-" : "+";
15  });
16});
```

Różne

```

1  Jedno odwołanie
2  function deleteWorkout(id, button) {
3      console.log(id);
4      const dane = {
5          id: id
6      };
7      const url = "/Trening/Delete/${id}";
8      fetch(url, {
9          method: 'POST',
10         headers: {
11             'Content-Type': 'application/json',
12             'X-Requested-With': 'XMLHttpRequest'
13         },
14         body: id
15     })
16     .then(response => {
17         if (response.ok) {
18             const row = button.closest('tr');
19             if (row) row.remove();
20         }
21     })
22     .catch(error => console.error('Błąd w żądaniu:', error));

```

AplikacjaTrenowania

```

10    private readonly ApplicationDbContext _context;
11    Odwołania: 0
12    public TreningController(ApplicationDbContext context)
13    {
14        _context = context;
15    }
16    private readonly ILogger<TreningController> _logger;
17    private static List<Trening> zapisaneTreninki = new List<Trening>();
18    private readonly Dictionary<string, List<string>> cwiczenia = new()
19    {
20        { "Siłowy", new List<string> { "Przysiadły", "Martwy ciąg", "Wyciskanie sztangi" } },
21        { "Kardio", new List<string> { "Bieganie", "Rowing", "Szakanka" } },
22        { "Stretching", new List<string> { "Joga", "Rozciąganie dynamiczne", "Pilates" } }
23    };
24    // GET: Trening
25    Odwołania: 0
26    public ActionResult Index()
27    {
28        var trening = _context.Trening.Include(x => x.Serie);
29        var daty = trening.Select(x => x.Data).Distinct().ToList();
30        ViewBag.Daty = daty;
31        return View(trening);
32    }
33    // GET: Trening/Details/5
34    Odwołania: 0
35    public ActionResult Details(int id) => View();
36    [HttpPost]
37    [Route("Trening/Delete/{id}")]
38    Odwołania: 0
39    public void Delete([FromBody] int id)
40    {
41        // var id = dane.id;
42        var trening = _context.Trening.Find(id);
43        if(trening != null) _context.Trening.Remove(trening);
44        _context.SaveChanges();
45    }
46    // GET: Trening/Create
47    Odwołania: 0
48    public ActionResult Create()
49    {
50        ViewBag.Cwiczenia = cwiczenia;
51        return View(new Trening());
52    }
53    // POST: Trening/Create
54    [HttpPost]
55    Odwołania: 0
56    public ActionResult Create(Trening model) {
57        _context.Trening.Add(model);
58        _context.SaveChanges();
59        return RedirectToAction("Save", new{id=model.IdTreningu});
60    }
61    // GET: Trening/Save
62    Odwołania: 0
63    public ActionResult Save(int id)
64    {
65        var trening = _context.Trening.Include(x => x.Serie).FirstOrDefault(x => x.IdTreningu == id);
66        return View(trening);
67    }

```

08.01.2025

ApplicationDbContext.cs

```
AplikacjaTrenowania
1  using Microsoft.AspNetCore.Identity.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore;
3  using AplikacjaTrenowania.Models;
4  using Microsoft.EntityFrameworkCore.Metadata.Builders;
5
6  namespace AplikacjaTrenowania.Areas.Identity.Data;
7
8  Odwołania: 9
9  public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
10 {
11     Odwołania: 0
12     public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)
13         : base(options)
14     {
15     }
16
17     Odwołania: 0
18     private class ApplicationUserEntityTypeConfiguration : IEntityTypeConfiguration<ApplicationUser>
19     {
20         Odwołania: 0
21         public void Configure(EntityTypeBuilder<ApplicationUser> builder)
22         {
23             builder.Property(x => x.FirstName).HasMaxLength(255);
24             builder.Property(x => x.LastName).HasMaxLength(255);
25         }
26     }
27
28     Odwołania: 5
29     public DbSet<Trening> Trening { get; set; }
30     Odwołania: 0
31     public DbSet<Seria> Seria { get; set; }
32 }
```

ApplicationUser.cs

```
AplikacjaTrenowania
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using Microsoft.AspNetCore.Identity;
6
7  namespace AplikacjaTrenowania.Areas.Identity.Data;
8
9  // Add profile data for application users by adding properties to the ApplicationUser class
10 Odwołania: 28
11 public class ApplicationUser : IdentityUser
12 {
13     Odwołania: 4
14     public string FirstName { get; set; }
15     Odwołania: 4
16     public string LastName { get; set; }
17 }
```

```
Logout.cshtml.cs  ✘ X ApplicationUser.cs      ApplicationDbContext.cs
AplikacjaTrenowania
1 // Licensed to the .NET Foundation under one or more agreements.
2 // The .NET Foundation licenses this file to you under the MIT license.
3 #nullable disable
4
5 using System;
6 using System.Threading.Tasks;
7 using Microsoft.AspNetCore.Authorization;
8 using AplikacjaTrenowania.Areas.Identity.Data;
9 using Microsoft.AspNetCore.Identity;
10 using Microsoft.AspNetCore.Mvc;
11 using Microsoft.AspNetCore.Mvc.RazorPages;
12 using Microsoft.Extensions.Logging;
13
14 namespace AplikacjaTrenowania.Areas.Identity.Pages.Account
15 {
16     public class LogoutModel : PageModel
17     {
18         private readonly SignInManager<ApplicationUser> _signInManager;
19         private readonly ILogger<LogoutModel> _logger;
20
21         public LogoutModel(SignInManager<ApplicationUser> signInManager, ILogger<LogoutModel> logger)
22         {
23             _signInManager = signInManager;
24             _logger = logger;
25         }
26
27         public async Task<IActionResult> OnPostAsync(string returnUrl = null)
28         {
29             await _signInManager.SignOutAsync();
30             _logger.LogInformation("User logged out.");
31             if (returnUrl != null)
32             {
33                 return LocalRedirect(returnUrl);
34             }
35             else
36             {
37                 // This needs to be a redirect so that the browser performs a new
38                 // request and the identity for the user gets updated.
39                 return RedirectToPage();
40             }
41         }
42     }
43 }
44
```

```
addserie.js    forms.js    Create.cshtml    TreningController.cs    expandingist.js    deleteworkout.js    Register.cshtml
Logout.cshtml.cs    ApplicationUser.cs    ApplicationDBContext.cs

AplikacjaTrenowania
8
9     <div class="row">
10        <div class="col-md-4">
11            <form id="registerForm" asp-route-returnUrl="@Model.ReturnUrl" method="post">
12                <h2>Create a new account.</h2>
13                <hr />
14                <div class="asp-validation-summary ModelOnly" class="text-danger" role="alert"></div>
15                <div class="form-floating mb-3">
16                    <input asp-for="Input.FirstName" class="form-control" autocomplete="username" aria-required="true" placeholder="Imię" />
17                    <label asp-for="Input.FirstName">First Name</label>
18                    <span asp-validation-for="Input.FirstName" class="text-danger"></span>
19                </div>
20                <div class="form-floating mb-3">
21                    <input asp-for="Input.LastName" class="form-control" autocomplete="username" aria-required="true" placeholder="Nazwisko" />
22                    <label asp-for="Input.LastName">Last Name</label>
23                    <span asp-validation-for="Input.LastName" class="text-danger"></span>
24                </div>
25                <div class="form-floating mb-3">
26                    <input asp-for="Input.Email" class="form-control" autocomplete="username" aria-required="true" placeholder="name@example.com" />
27                    <label asp-for="Input.Email">Email</label>
28                    <span asp-validation-for="Input.Email" class="text-danger"></span>
29                </div>
30                <div class="form-floating mb-3">
31                    <input asp-for="Input.Password" class="form-control" autocomplete="new-password" aria-required="true" placeholder="password" />
32                    <label asp-for="Input.Password">Password</label>
33                    <span asp-validation-for="Input.Password" class="text-danger"></span>
34                </div>
35                <div class="form-floating mb-3">
36                    <input asp-for="Input.ConfirmPassword" class="form-control" autocomplete="new-password" aria-required="true" placeholder="password" />
37                    <label asp-for="Input.ConfirmPassword">Confirm Password</label>
38                    <span asp-validation-for="Input.ConfirmPassword" class="text-danger"></span>
39                </div>
40                <button id="registerSubmit" type="submit" class="w-100 btn btn-lg btn-primary">Register</button>
41            </form>
42        </div>
43        <div class="col-md-6 col-md-offset-2">
44            <section>
45                <h3>Use another service to register.</h3>
46                <hr />
47                @{
48                    if ((Model.ExternalLogins?.Count ?? 0) == 0)
49                    {
50                        <div>
51                            <p>There are no external authentication services configured. See this <a href="https://go.microsoft.com/fwlink/?LinkID=532715">article</a> about Setting up this ASP.NET application to support logging in via external services.</p>
52                        </div>
53                    }
54                    else
55                    {
56                        <form id="external-account" asp-page="./ExternalLogin" asp-route-returnUrl="@Model.ReturnUrl" method="post" class="form-horizontal">
57                            <div>
58                                <p>
59                                    #foreach (var provider in Model.ExternalLogins)
60                                    {
61                                        <button type="submit" class="btn btn-primary" name="provider" value="@provider.Name" title="Log in using your @provider.DisplayName account">@provider.DisplayName</button>
62                                    }
63                                </p>
64                            </div>
65                        </form>
66                    }
67                }
68            </section>
69        </div>
70    </div>
71    <section Scripts >
72        <partial name="_ValidationScriptsPartial" />
73    </section>
74
```

```
Register.cshtml.cs ✘ X Logout.cshtml.cs ApplicationUser.cs ApplicationDbContext.cs
ApplikacjaTrenowania
1 // Licensed to the .NET Foundation under one or more agreements.
2 // The .NET Foundation licenses this file to you under the MIT license.
3 #nullable disable
4
5 using System;
6 using System.Collections.Generic;
7 using System.ComponentModel.DataAnnotations;
8 using System.Linq;
9 using System.Text;
10 using System.Text.Encodings.Web;
11 using System.Threading;
12 using System.Threading.Tasks;
13 using Microsoft.AspNetCore.Authentication;
14 using Microsoft.AspNetCore.Authorization;
15 using ApplikacjaTrenowania.Areas.Identity.Data;
16 using Microsoft.AspNetCore.Identity;
17 using Microsoft.AspNetCore.Identity.UI.Services;
18 using Microsoft.AspNetCore.Mvc;
19 using Microsoft.AspNetCore.Mvc.RazorPages;
20 using Microsoft.AspNetCore.WebUtilities;
21 using Microsoft.Extensions.Logging;
22
23 namespace ApplikacjaTrenowania.Areas.Identity.Pages.Account
24 {
25     Odwołania: 8
26     public class RegisterModel : PageModel
27     {
28         private readonly SignInManager<ApplicationUser> _signInManager;
29         private readonly UserManager<ApplicationUser> _userManager;
30         private readonly IUserStore<ApplicationUser> _userStore;
31         private readonly IUserEmailStore<ApplicationUser> _emailStore;
32         private readonly ILogger<RegisterModel> _logger;
33         private readonly IEmailSender _emailSender;
34
35         public RegisterModel(
36             UserManager< ApplicationUser > userManager,
37             IUserStore< ApplicationUser > userStore,
38             SignInManager< ApplicationUser > signInManager,
39             ILogger< RegisterModel > logger,
40             IEmailSender emailSender)
41         {
42             _userManager = userManager;
43             _userStore = userStore;
44             _emailStore = GetEmailStore();
45             _signInManager = signInManager;
46             _logger = logger;
47             _emailSender = emailSender;
48         }
49
50         /// <summary>
51         ///     This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
52         ///     directly from your code. This API may change or be removed in future releases.
53         /// </summary>
54         [BindProperty]
55         Odwołania: 21
56         public InputModel Input { get; set; }
57
58         /// <summary>
59         ///     This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
60         ///     directly from your code. This API may change or be removed in future releases.
61         /// </summary>
62         Odwołania: 3
63         public string ReturnUrl { get; set; }
64
65         /// <summary>
66         ///     This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
67         ///     directly from your code. This API may change or be removed in future releases.
68         /// </summary>
69     }
70 }
```

```
ApplicationDB...elSnapshot.cs Register.cshtml.cs Logout.cshtml.cs ApplicationUser.cs ApplicationDbContext.cs
ApplikacjaTrenowania AplikacjaTrenowania.Areas.Identity.Pages.Account.RegisterModel

66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130

public IList<AuthenticationScheme> ExternalLogins { get; set; }

/// <summary>
/// This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
/// directly from your code. This API may change or be removed in future releases.
/// </summary>
[odwolania:4]
public class InputModel
{
    [Required]
    [Display(Name = "FirstName")]
    [StringLength(255, ErrorMessage = "max 255.")]
    Odwołanie:4
    public string FirstName { get; set; }

    [Required]
    [Display(Name = "LastName")]
    [StringLength(255, ErrorMessage = "max 255.")]
    Odwołanie:4
    public string LastName { get; set; }

    /// <summary>
    /// This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
    /// directly from your code. This API may change or be removed in future releases.
    /// </summary>
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    Odwołanie:6
    public string Email { get; set; }

    /// <summary>
    /// This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
    /// directly from your code. This API may change or be removed in future releases.
    /// </summary>
    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} and at max {1} characters long.", MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    Odwołanie:4
    public string Password { get; set; }

    /// <summary>
    /// This API supports the ASP.NET Core Identity default UI infrastructure and is not intended to be used
    /// directly from your code. This API may change or be removed in future releases.
    /// </summary>
    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not match.")] Odwołanie:3
    public string ConfirmPassword { get; set; }
}

Odwołanie:0
public async Task OnGetAsync(string returnUrl = null)
{
    returnUrl = returnUrl;
    ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
}

Odwołanie:0
public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl ??= Url.Content("~/");
    ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
    if (ModelState.IsValid)
    {
        var user = CreateUser();

        user.FirstName = Input.FirstName;
        user.LastName = Input.LastName;
        user.EmailConfirmed = true;

        await _userStore.SetUserNameAsync(user, Input.Email, CancellationToken.None);
    }
}

66 %
Nie znaleziono żadnych problemów
```

```
Register.cshtml.cs  Logout.cshtml.cs  ApplicationUser.cs  ApplicationDbContext.cs
AplikacjaTrenowania  AplikacjaTrenowania.Areas.Identity.Pages.Account.RegisterModel

151  await _emailStore.SetEmailAsync(user, Input.Email, CancellationToken.None);
152  var result = await _userManager.CreateAsync(user, Input.Password);
153
154  if (result.Succeeded)
155  {
156      //logger.LogInformation("User created a new account with password.");
157
158  //var userId = await _userManager.GetUserIdAsync(user);
159  //var code = await _userManager.GenerateEmailConfirmationTokenAsync(user);
160  //code = WebEncoders.Base64UrlEncode(Encoding.UTF8.GetBytes(code));
161  //var callbackUrl = Url.Page(
162  //    "/Account/ConfirmEmail",
163  //    pageHandler: null,
164  //    values: new { area = "Identity", userId = userId, code = code, returnUrl = returnUrl },
165  //    protocol: Request.Scheme);
166
167  //await _emailSender.SendEmailAsync(Input.Email, "Confirm your email",
168  //    $"Please confirm your account by <a href='{HtmlEncoder.Default.Encode(callbackUrl)}>clicking here</a>.");
169
170  if (_userManager.Options.SignIn.RequireConfirmedAccount)
171  {
172      return RedirectToPage("RegisterConfirmation", new { email = Input.Email, returnUrl = returnUrl });
173  }
174  else
175  {
176      await _signInManager.SignInAsync(user, isPersistent: false);
177      return LocalRedirect(returnUrl);
178  }
179
180  foreach (var error in result.Errors)
181  {
182      ModelState.AddModelError(string.Empty, error.Description);
183  }
184
185  // If we got this far, something failed, redisplay form
186  return Page();
187}
188
189 1 odwołanie
190  private ApplicationUser CreateUser()
191  {
192      try
193      {
194          return Activator.CreateInstance<ApplicationUser>();
195      }
196      catch
197      {
198          throw new InvalidOperationException($"Can't create an instance of '{nameof(ApplicationUser)}'. " +
199              $"Ensure that '{nameof(ApplicationUser)}' is not an abstract class and has a parameterless constructor, or alternatively " +
200              $"override the register page in /Areas/Identity/Pages/Account/Register.cshtml");
201      }
202
203  1 odwołanie
204  private IUserEmailStore<ApplicationUser> GetEmailStore()
205  {
206      if (!_userManager.SupportsUserEmail)
207      {
208          throw new NotSupportedException("The default UI requires a user store with email support.");
209      }
210      return (IUserEmailStore<ApplicationUser>) _userStore;
211  }
212
213  1 odwołanie
214
```

```
ApplicationD...elSnapshot.cs Register.cshtml.cs Logout.cshtml.cs ApplicationUser.cs ApplicationDbContext.cs
ApplikacjaTrenowania
1 // auto-generated /
2 using System;
3 using ApplikacjaTrenowania.Areas.Identity.Data;
4 using Microsoft.EntityFrameworkCore;
5 using Microsoft.EntityFrameworkCore.Infrastructure;
6 using Microsoft.EntityFrameworkCore.Metadata;
7 using Microsoft.EntityFrameworkCore.Storage.ValueConversion;
8
9 #nullable disable
10
11 namespace ApplikacjaTrenowania.Migrations
12 {
13     [DbContext(typeof(ApplicationDbContext))]
14     partial class ApplicationDBContextModelSnapshot : ModelSnapshot
15     {
16         Odwołania:0
17         protected override void BuildModel(ModelBuilder modelBuilder)
18         {
19             #pragma warning disable 612, 618
20             modelBuilder
21                 .HasAnnotation("ProductVersion", "8.0.11")
22                 .HasAnnotation("Relational:MaxIdentifierLength", 128);
23
24            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);
25
26             modelBuilder.Entity<ApplikacjaTrenowania.Areas.Identity.Data.ApplicationUser>, b =>
27             {
28                 b.Property<string>("Id")
29                     .HasColumnType("nvarchar(450)");
30
31                 b.Property<int>("AccessFailedCount")
32                     .HasColumnType("int");
33
34                 b.Property<string>("ConcurrencyStamp")
35                     .IsConcurrencyToken()
36                     .HasColumnType("nvarchar(max)");
37
38                 b.Property<string>("Email")
39                     .HasMaxLength(256)
40                     .HasColumnType("nvarchar(256)");
41
42                 b.Property<bool>("EmailConfirmed")
43                     .HasColumnType("bit");
44
45                 b.Property<string>("FirstName")
46                     .IsRequired()
47                     .HasColumnType("nvarchar(max)");
48
49                 b.Property<string>("LastName")
50                     .IsRequired()
51                     .HasColumnType("nvarchar(max)");
52
53                 b.Property<bool>("LockoutEnabled")
54                     .HasColumnType("bit");
55
56                 b.Property<DateTimeOffset?>("LockoutEnd")
57                     .HasColumnType("datetimeoffset");
58
59                 b.Property<string>("NormalizedEmail")
60                     .HasMaxLength(256)
61                     .HasColumnType("nvarchar(256)");
62
63                 b.Property<string>("NormalizedUserName")
64                     .HasMaxLength(256)
65                     .HasColumnType("nvarchar(256)");
66
67                 b.Property<string>("PasswordHash")
68                     .HasColumnType("nvarchar(max)");
69
70                 b.Property<string>("PhoneNumber")
71                     .HasColumnType("nvarchar(max)");
72
73             };
74         };
75     }
76 }
```



```
ApplicationD...elSnapshot.cs Register.cshtml.cs Logout.cshtml.cs ApplicationUser.cs ApplicationDbContext.cs
ApplikacjaTrenowania
143     .IsRequired()
144     .HasColumnType("nvarchar(100)");
145 
146     b.HasKey("IdTreningu");
147 
148     b.ToTable("Trening");
149 }
150 
151 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityRole", b =>
152 {
153     b.Property<"string">("Id")
154     .HasColumnType("nvarchar(450)");
155 
156     b.Property<"string">("ConcurrencyStamp")
157     .IsConcurrencyToken()
158     .HasColumnType("nvarchar(max)");
159 
160     b.Property<"string">("Name")
161     .HasMaxLength(256)
162     .HasColumnType("nvarchar(256)");
163 
164     b.Property<"string">("Normalized Name")
165     .HasMaxLength(256)
166     .HasColumnType("nvarchar(256)");
167 
168     b.HasKey("Id");
169 
170     b.HasIndex("Normalized Name")
171     .IsUnique()
172     .HasDatabaseName("RoleNameIndex")
173     .HasFilter("[Normalized Name] IS NOT NULL");
174 
175     b.ToTable("AspNetRoles", (string)null);
176 }
177 
178 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityRoleClaim<"string">, b =>
179 {
180     b.Property<"int">("Id")
181     .ValueGeneratedOnAdd()
182     .HasColumnType("int");
183 
184    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<"int">("Id"));
185 
186     b.Property<"string">("ClaimType")
187     .HasColumnType("nvarchar(max)");
188 
189     b.Property<"string">("ClaimValue")
190     .HasColumnType("nvarchar(max)");
191 
192     b.Property<"string">("RoleId")
193     .IsRequired()
194     .HasColumnType("nvarchar(450)");
195 
196     b.HasKey("Id");
197 
198     b.HasIndex("RoleId");
199 
200     b.ToTable("AspNetRoleClaims", (string)null);
201 }
202 
203 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityUserClaim<"string">, b =>
204 {
205     b.Property<"int">("Id")
206     .ValueGeneratedOnAdd()
207     .HasColumnType("int");
208 
209    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<"int">("Id"));
210 
211     b.Property<"string">("ClaimType")
212     .HasColumnType("nvarchar(max)");
213 
214     b.Property<"string">("ClaimValue")
215     .HasColumnType("nvarchar(max)");
66 % Nie znaleziono żadnych problemów
```

```
ApplicationD...elSnapshot.cs Register.cshtml.cs Logout.cshtml.cs ApplicationUser.cs ApplicationDBContext.cs
ApplikacjaTrenowania
216
217     b.Property<string>("UserId")
218         .IsRequired()
219         .HasColumnType("nvarchar(450)");
220
221     b.HasKey("Id");
222
223     b.HasIndex("UserId");
224
225     b.ToTable("AspNetUserClaims", (string)null);
226
227 });
228
229 modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserLogin<string>", b =>
230 {
231     b.Property<string>("LoginProvider")
232         .HasMaxLength(128)
233         .HasColumnType("nvarchar(128)");
234
235     b.Property<string>("ProviderKey")
236         .HasMaxLength(128)
237         .HasColumnType("nvarchar(128)");
238
239     b.Property<string>("ProviderDisplayName")
240         .HasColumnType("nvarchar(max)");
241
242     b.Property<string>("UserId")
243         .IsRequired()
244         .HasColumnType("nvarchar(450)");
245
246     b.HasKey("LoginProvider", "ProviderKey");
247
248     b.HasIndex("UserId");
249
250     b.ToTable("AspNetUserLogins", (string)null);
251 });
252
253 modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserRole<string>", b =>
254 {
255     b.Property<string>("UserId")
256         .HasColumnType("nvarchar(450)");
257
258     b.Property<string>("RoleId")
259         .HasColumnType("nvarchar(450)");
260
261     b.HasKey("UserId", "RoleId");
262
263     b.HasIndex("RoleId");
264
265     b.ToTable("AspNetUserRoles", (string)null);
266 });
267
268 modelBuilder.Entity("Microsoft.AspNetCore.Identity.IdentityUserToken<string>", b =>
269 {
270     b.Property<string>("UserId")
271         .HasColumnType("nvarchar(450)");
272
273     b.Property<string>("LoginProvider")
274         .HasMaxLength(128)
275         .HasColumnType("nvarchar(128)");
276
277     b.Property<string>("Name")
278         .HasMaxLength(128)
279         .HasColumnType("nvarchar(128)");
280
281     b.Property<string>("Value")
282         .HasColumnType("nvarchar(max)");
283
284     b.HasKey("UserId", "LoginProvider", "Name");
285
286     b.ToTable("AspNetUserTokens", (string)null);
287 });
288
289 modelBuilder.Entity("ApplikacjaTrenowania.Models.Seria", b =>
```

```
ApplicationD...elSnapshot.cs Register.cshtml.cs Logout.cshtml.cs ApplicationUser.cs ApplicationDbContext.cs
ApplikacjaTrenowania
289     {
290         b.HasOne("ApplikacjaTrenowania.Models.Trening", "Trening")
291             .WithMany("Serie")
292             .HasForeignKey("idTreningu")
293             .OnDelete(DeleteBehavior.Cascade)
294             .IsRequired();
295
296         b.Navigation("Trening");
297     });
298
299 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityRoleClaim<string>">, b =>
300 {
301     b.HasOne("Microsoft.AspNetCore.Identity.IdentityRole", null)
302         .WithMany()
303         .HasForeignKey("RoleId")
304         .OnDelete(DeleteBehavior.Cascade)
305         .IsRequired();
306 };
307
308 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityUserClaim<string>">, b =>
309 {
310     b.HasOne("ApplikacjaTrenowania.Areas.Identity.Data.ApplicationUser", null)
311         .WithMany()
312         .HasForeignKey("UserId")
313         .OnDelete(DeleteBehavior.Cascade)
314         .IsRequired();
315 };
316
317 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityUserLogin<string>">, b =>
318 {
319     b.HasOne("ApplikacjaTrenowania.Areas.Identity.Data ApplicationUser", null)
320         .WithMany()
321         .HasForeignKey("UserId")
322         .OnDelete(DeleteBehavior.Cascade)
323         .IsRequired();
324 };
325
326 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityUserRole<string>">, b =>
327 {
328     b.HasOne("Microsoft.AspNetCore.Identity.IdentityRole", null)
329         .WithMany()
330         .HasForeignKey("RoleId")
331         .OnDelete(DeleteBehavior.Cascade)
332         .IsRequired();
333
334     b.HasOne("ApplikacjaTrenowania.Areas.Identity.Data ApplicationUser", null)
335         .WithMany()
336         .HasForeignKey("UserId")
337         .OnDelete(DeleteBehavior.Cascade)
338         .IsRequired();
339 };
340
341 modelBuilder.Entity<"Microsoft.AspNetCore.Identity.IdentityUserToken<string>">, b =>
342 {
343     b.HasOne("ApplikacjaTrenowania.Areas.Identity.Data ApplicationUser", null)
344         .WithMany()
345         .HasForeignKey("UserId")
346         .OnDelete(DeleteBehavior.Cascade)
347         .IsRequired();
348 };
349
350 modelBuilder.Entity<"ApplikacjaTrenowania.Models.Trening">, b =>
351 {
352     b.Navigation("Serie");
353 };
354 #pragma warning restore 612, 618
355 }
356 }
357 }
```

```
Layout.cshtml  ApplicationDB...elSnapshot.cs  Register.cshtml.cs  Logout.cshtml.cs  ApplicationUser.cs  ApplicationDbContext.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" aria-expanded="false" aria-label="Toggle navigation">
17                     <span class="navbar-toggler-icon"></span>
18                 </button>
19                 <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
20                     <ul class="navbar-nav flex-grow-1">
21                         <li class="nav-item">
22                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Index">Trening</a>
23                         </li>
24                         <li class="nav-item">
25                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Woda">Woda</a>
26                         </li>
27                         <li class="nav-item">
28                             <a class="nav-link text-dark" asp-area="" asp-controller="Trening" asp-action="Biatko">Biatko</a>
29                         </li>
30                     </ul>
31                 </div>
32             <partial name="_LoginPartial" />
33         </div>
34     </header>
35     <div class="container">
36         <main role="main" class="pb-3">
37             @RenderBody()
38         </main>
39     </div>
40 
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     </footer>
46     <script src="/lib/jquery/dist/jquery.min.js"></script>
47     <script src="/lib/bootstrap/dist/js/bootstrap.bundle.min.js"></script>
48     <script src="~/js/site.js" asp-append-version="true"></script>
49     @await RenderSectionAsync("Scripts", required: false)
50 </body>
51 </HTML>
```

```
Program.cs  Layout.cshtml  ApplicationDB...elSnapshot.cs  Register.cshtml.cs  Logout.cshtml.cs  ApplicationUser.cs  ApplicationDbContext.cs
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 =>
10    {
11        options.SignIn.RequireConfirmedAccount = true;
12        options.SignIn.RequireConfirmedEmail = false;
13    })
14    .AddRoles<IdentityRole>()
15    .AddEntityFrameworkStores<ApplicationDbContext>();
16 
17  builder.Services.ConfigureApplicationCookie(options =>
18    {
19        options.LogoutPath = "/Account/Logout";
20    });
21 
22 // Add services to the container.
23 builder.Services.AddControllersWithViews();
24 
25 var app = builder.Build();
26 
27 using (var scope = app.Services.CreateScope())
28 {
29     var userManager = scope.ServiceProvider.GetRequiredService<UserManager<ApplicationUser>>();
30     var roleManager = scope.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>();
31 
32     var roleExists = await roleManager.RoleExistsAsync("Admin");
33     if (!roleExists)
34     {
35         var role = new IdentityRole("Admin");
36         await roleManager.CreateAsync(role);
37     }
38 
39     var user = await userManager.FindByEmailAsync("admin@admin.com");
40     if (user == null)
41     {
42         user = new ApplicationUser { FirstName = "Marcin", LastName = "Tomczyk", UserName = "admin@admin.com", Email = "admin@admin.com", EmailConfirmed = true };
43         await userManager.CreateAsync(user, "załóż55");
44     }
45 
46     if (await userManager.IsInRoleAsync(user, "Admin"))
47     {
48         await userManager.AddToRoleAsync(user, "Admin");
49     }
50 
51     var otherUser = await userManager.FindByEmailAsync("user@user.com");
52     if (otherUser == null)
53     {
54         otherUser = new ApplicationUser { FirstName = "Kacper", LastName = "Nowak", UserName = "user@user.com", Email = "user@user.com", EmailConfirmed = true };
55         await userManager.CreateAsync(otherUser, "załóż55");
56     }
57 }
```

```
Program.cs  X  _Layout.cshtml  ApplicationDB...elSnapshot.cs  Register.cshtml.cs  Logout.cshtml.cs  ApplicationUser.cs  ApplicationDbContext.cs
ApplikacjaTrenowania
38
39     var user = await userManager.FindByEmailAsync("admin@admin.com");
40     if (user == null)
41     {
42         user = new ApplicationUser { FirstName = "Marcin", LastName = "Tomczyk", UserName = "admin@admin.com", Email = "admin@admin.com", EmailConfirmed = true};
43         await userManager.CreateAsync(user, "zaql@WSX");
44     }
45
46     if (!await userManager.IsInRoleAsync(user, "Admin"))
47     {
48         await userManager.AddToRoleAsync(user, "Admin");
49     }
50
51     var otherUser = await userManager.FindByEmailAsync("user@user.com");
52     if (otherUser == null)
53     {
54         otherUser = new ApplicationUser { FirstName = "Kacper", LastName = "Nowak", UserName = "user@user.com", Email = "user@user.com", EmailConfirmed = true};
55         await userManager.CreateAsync(otherUser, "zaql@WSX");
56     }
57
58     // Configure the HTTP request pipeline.
59     if (app.Environment.IsDevelopment())
60     {
61         app.UseExceptionHandler("/Home/Error");
62         // The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.
63         app.UseHsts();
64     }
65
66     app.UseHttpsRedirection();
67     app.UseStaticFiles();
68
69     app.UseRouting();
70
71     app.UseAuthentication();
72
73     app.UseAuthorization();
74
75     app.MapControllerRoute(
76         name: "default",
77         pattern: "{controller=Home}/{action=Index}/{id?}");
78
79     app.MapRazorPages();
80
81     app.Run();
82
83
```