

20.12.2024

## Sieć Web i chmura (4)



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

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

mvc

Wyczyść wszystko

Wszystkie języki

Wszystkie platformy

Wszystkie typy projektów



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

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

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

## Konfiguruj nowy projekt

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

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

Nazwa projektu

AplikacjaTrenowania

Lokalizacja

F:\

Nazwa rozwiązania ⓘ

AplikacjaTrenowania

☐ Umieść rozwiązanie i projekt w tym samym katalogu

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

## Informacje dodatkowe

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

C#

Linux

macOS

Windows

Chmura

Usługa

Internet

Platforma ⓘ

.NET 8.0 (Długoterminowa pomoc techniczna)

Typ uwierzytelniania ⓘ

Brak

☒ Konfiguruj dla protokołu HTTPS ⓘ

☐ Włączanie obsługi kontenerów ⓘ

System operacyjny kontenera ⓘ

Linux

Typ kompilacji kontenera ⓘ

Dockerfile

☐ Nie używaj instrukcji najwyższego poziomu ⓘ

☐ Rejestracja w orkiestracji .NET Aspire ⓘ

Wersja Aspire ⓘ

9.0

## Dodaj nowy element szkieletowy

▲ Zainstalowane

Tożsamość

Układ

▲ Wspólne

API

▸ Blazor

▸ MVC

Strony Razor



Tożsamość

**Tożsamość**

przez Microsoft  
v1.0.0.0

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

Identyfikator: IdentityScaffolder

Dodaj

Anuluj

## Dodaj Tożsamość

Wybierz istniejącą stronę układu lub określ nową:

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

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

☐ Zastąp wszystkie pliki

Wybierz pliki do zastąpienia

☐ Account\StatusMessage

☐ Account\ConfirmEmailChange

☐ Account\ForgotPasswordConfirmation

☐ Account>LoginWith2fa

☐ Account\Manage\Layout

☐ Account\Manage\ChangePassword

☐ Account\Manage\DownloadPersonalData

☐ Account\Manage\ExternalLogins

☐ Account\Manage\PersonalData

☐ Account\Manage\ShowRecoveryCodes

☐ Account\RegisterConfirmation

☐ Account\ResetPasswordConfirmation

☐ Account\AccessDenied

☐ Account\ExternalLogin

☐ Account\Lockout

☐ Account>LoginWithRecoveryCode

☐ Account\Manage\ManageNav

☐ Account\Manage\DeletePersonalData

☐ Account\Manage\Email

☐ Account\Manage\GenerateRecoveryCodes

☐ Account\Manage\ResetAuthenticator

☐ Account\Manage\TwoFactorAuthentication

☐ Account\ResendEmailConfirmation

☐ Account\ConfirmEmail

☐ Account\ForgotPassword

☒ Account>Login

☒ Account\Logout

☐ Account\Manage\StatusMessage

☐ Account\Manage\Disable2fa

☐ Account\Manage\EnableAuthenticator

☐ Account\Manage\Index

☐ Account\Manage\SetPassword

☒ Account\Register

☐ Account\ResetPassword

Klasa DbContext

ApplicationDbContext



Dostawca bazy danych

SQL Server



Klasa użytkownika

ApplicationUser



Dodaj

Anuluj

# SQL Server

Login Connection Properties Always Encrypted Additional Connection Parameters

## Server

Server type: Database Engine

Server name: DESKTOP-TFJR5RH

Authentication: Windows Authentication

User name: DESKTOP-TFJR5RH\Damian

Password:

☐ Remember password

## Connection Security

Encryption: Optional

☐ Trust server certificate

Host name in certificate:

Connect

Cancel

Help

Options <<

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

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

8.0.11  
9.0.0

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

8.0.11  
9.0.0

Tworzenie repozytorium Git

Wypchnij do nowego zdalnego

GitHub

Azure DevOps

Inne

Istniejące zdalne

Tylko lokalne

Zainicjuj lokalne repozytorium Git

Ścieżka lokalna

F:\AplikacjaTrenowania

Szablon .gitignore

Domyślne (VisualStudio)

Szablon licencji

Brak

☐ Dodaj README.md

Utwórz nowe repozytorium GitHub

Konto

DamianSikoraUP (GitHub)

Właściciel

DamianSikoraUP

Nazwa repozytorium

AplikacjaTrenowania

Opis

Wprowadź opis repozytorium GitHub <Opcjonalnie>

Widoczność

Private

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

Przekaż kod do usługi GitHub

https://github.com/DamianSikoraUP/AplikacjaTrenowania

Utwórz i wypchnij

Anuluj

General

Access

Collaborators

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Codespaces

Pages

Security

Code security

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Who has access

Private repository

Only those with access to this repository can view it

Manage

PRIVATE REPOSITORY

Only those with access to this repository can view it.

Manage

DIRECT ACCESS

1 user has access to this repository.

1 collaborator.

Manage access

Add people

Select all

Type

Find a collaborator...

DawidBilUP

Collaborator

Previous

Next

22.12.2024

Przygotowaliśmy początkowe szkice które miały za zadanie rozplanować działanie aplikacji

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

DATA GODZINA

Rodzaj treningu LISTA

Wybierz ćwiczenia Lista

Kłosa 1.  kg /  powtórzeń

Dodaj Serię

Zapisz

Zapisz Ćwiczenie

Zapisane :



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

DATA

Dodaj wypitą wodę (ml)

Zapisz

Wypita woda

----- ml

Dzienny cel wody

----- ml

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

DATA

Dobry Zjedzone białko (g)



Zapisz

Zjedzone białko

----- g

Dzienny cel białka

----- g

HomeController.cs\* [icon] X

AplikacjaTrenowania [icon] AplikacjaTrenowania.C

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

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

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

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

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

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

Odwołania: 0

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

Odwołania: 0

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

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



## Dodaj nowy element szkieletowy

### ▲ Zainstalowane

Tożsamość  
Układ

### ▲ Wspólne

API

▸ Blazor

### ▲ MVC

Kontroler

Wyświetl

Strony Razor



Widok Razor — pusty



Widok Razor

### Widok Razor

przez Microsoft  
v1.0.0.0

Widok Razor z silnie typizowanym  
modelem lub bez niego

Identyfikator: RazorViewScaffolder

Dodaj

Anuluj

## Dodaj Widok Razor

Nazwa widoku

Trening

Szablon

Empty (bez modelu)

Klasa modelu

Klasa DbContext

Dostawca bazy danych

### Opcje



Utwórz jako widok częściowy



Odwołaj się do bibliotek skryptów

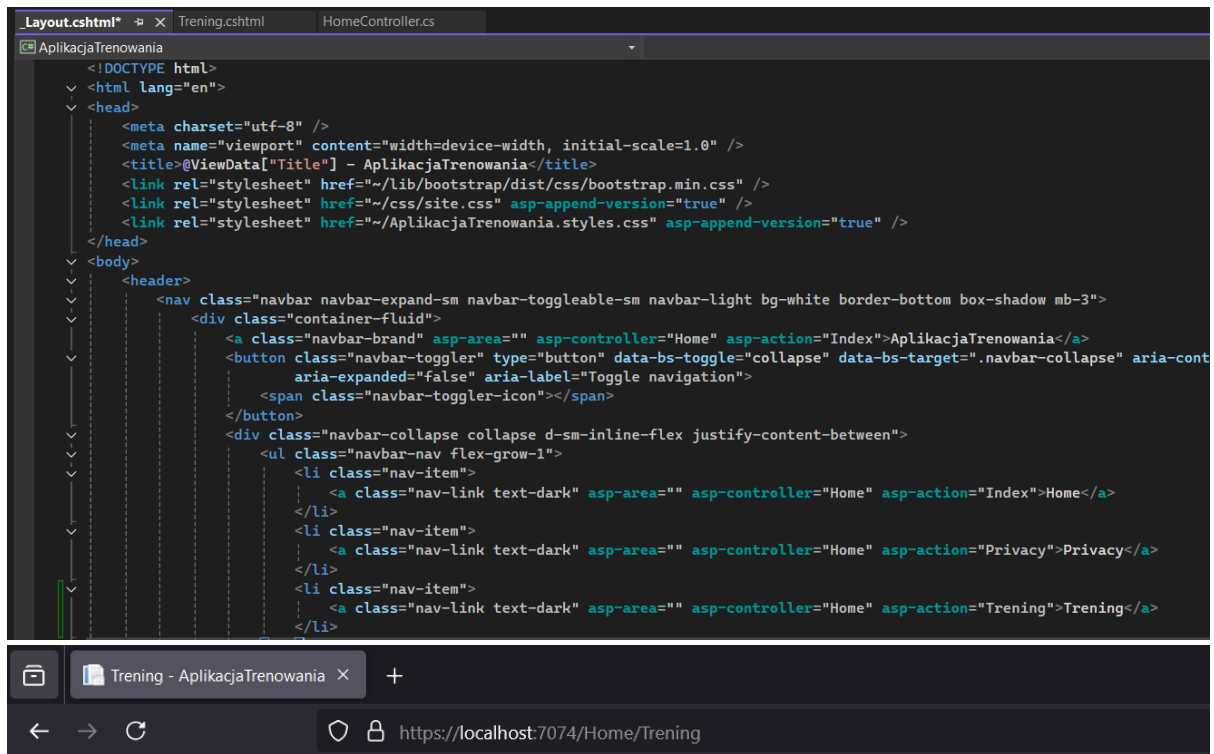


Użyj strony układu

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

Dodaj

Anuluj

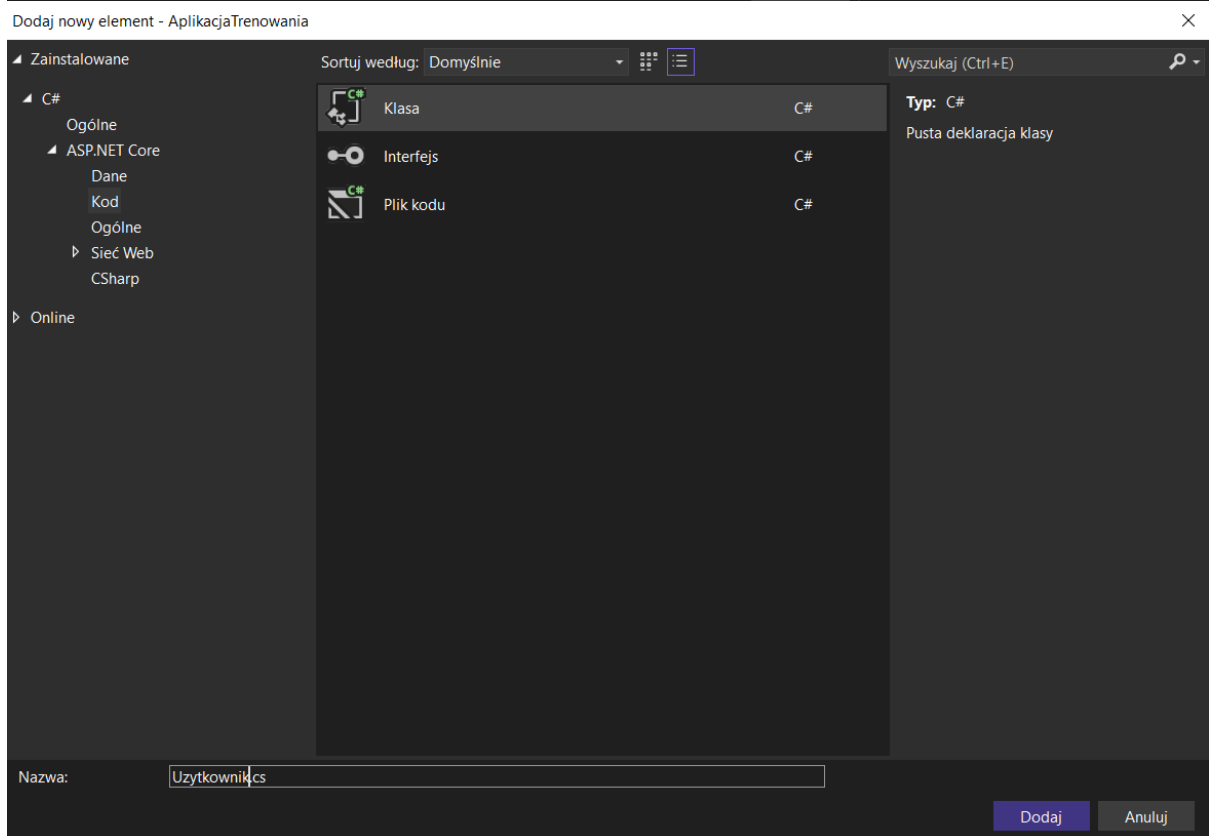
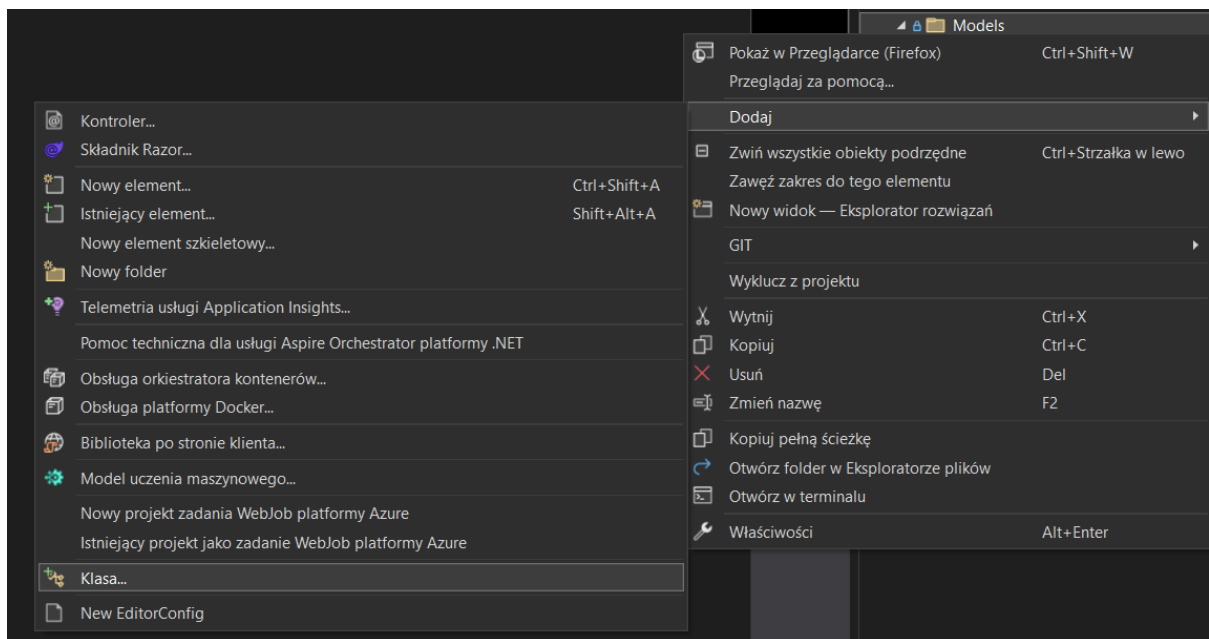


AplikacjaTrenowania   Home   Privacy   Trening

## Trening

Analogicznie dodaliśmy podstrony Woda i Białko

AplikacjaTrenowania   Home   Privacy   Trening   Woda   Białko



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