

# Tytuł projektu: System zarządzania dietą i ćwiczeniami

Autorzy: Krzysztof Jasiak, Kordian Kosmala

## Krótki opis działania projektu:

Projekt jest prostym systemem, który umożliwia użytkownikom rejestrację, logowanie i zarządzanie swoimi codziennymi kaloriami. Użytkownicy mogą dodawać posiłki, przypisywać im kalorie i składniki, a także wielokrotnie wykorzystywać je w różnych dniach. Dodatkowo, system umożliwia dodawanie ćwiczeń do wybranego dnia. Wszystkie dane są przechowywane w bazie danych i powiązane z kontem użytkownika.

## Instrukcje pierwszego uruchomienia projektu

### 1. Klonowanie repozytorium:

Sklonuj projekt z repozytorium (np. GitHub lub GitLab) na swój lokalny komputer lub pobierz .zip z projektem i wypakuj go w folderze z projektami VS

### 2. Baza danych:

Kolejnym elementem jest aktualizacja bazy danych, w tym celu musimy otworzyć **konsole menadżera pakietów NuGet** i wpisać polecenie *Update-Database*.

### 3. Uruchomienie projektu:

- W Visual Studio ustaw projekt jako startowy i uruchom aplikację (klawisz **F5**).
- Przeglądarka powinna otworzyć aplikację pod adresem <https://localhost:7188>. (adres w zależności od ustawień)

### 4. Tworzenie konta użytkownika:

- Zarejestruj się za pomocą strony rejestracji, aby rozpocząć korzystanie z aplikacji.

## Opis struktury projektu:

Projekt jest zbudowany w technologii **C#** z użyciem **.NET 8**. Bazą danych jest **MS SQL Server**. Struktura aplikacji składa się z następujących głównych komponentów:

- **Modele** – reprezentują dane w systemie, takie jak użytkownicy, posiłki, ćwiczenia, kalorie, itp.
- **Kontrolery** – odpowiadają za logikę aplikacji i interakcję z użytkownikiem, w tym rejestrację, logowanie, dodawanie posiłków, ćwiczeń, itp.
- **Widoki** – zawierają interfejs użytkownika, umożliwiające dodawanie danych i wyświetlanie statystyk.
- **Baza danych** – przechowuje informacje o użytkownikach, posiłkach, ćwiczeniach i dziennych kaloriach.
- **Logowanie i rejestracja użytkowników** – implementowane za pomocą Microsoft Identity w stronach ASP.NET, umożliwiające użytkownikom tworzenie kont, logowanie i zarządzanie swoimi danymi.

Projekt posiada również część wizualną, pozwalającą użytkownikowi na wygodne dodawanie danych i śledzenie postępów.

## Wylistowane modele

Wszystkie modele zostały zaimplementowane przy użyciu Entity Framework, co pozwala na mapowanie ich na tabele w bazie danych.

### 1. Dashboard

Reprezentuje podsumowanie dnia użytkownika, np. bilans kalorii.

#### o Atrybuty:

```
public int Id { get; set; }
public string UserName { get; set; } = string.Empty;
public int TotalCalories { get; set; }
public int TotalExercises { get; set; }
public int TotalBurnedCalories { get; set; }
public int CaloriesBalance { get; set; }
public IEnumerable<MealsList> Meals { get; set; } = new List<MealsList>();
public IEnumerable<ExercisesList> Exercises { get; set; } = new
List<ExercisesList>();
```

### 2. ExercisesList

Przechowuje dane dotyczące ćwiczeń użytkownika.

- **Atrybuty:**  
public int Id { get; set; }  
[Required]  
[Display(Name = "Nazwa ćwiczenia")]  
public string Name { get; set; }  
[Required]  
[Display(Name = "Spalone kalorie")]  
public int BurnedCalories { get; set; }  
[Required]  
[Display(Name = "Data wykonania ćwiczenia")]  
public DateTime TrainingAt { get; set; }  
public string? UserId { get; set; }  
public IdentityUser? User { get; set; }  
public double CaloriesBurned { get; set; }

### 3. MealsList

Zawiera informacje o posiłkach użytkownika.

- **Atrybuty:**  
public int Id { get; set; }  
[Required]  
[Display(Name = "Nazwa posiłku")]  
public string Name { get; set; }  
[Required]  
[Display(Name = "Kalorie")]  
public int Calories { get; set; }  
[Display(Name = "Opis")]  
public string? Description { get; set; }  
[Required]  
[Display(Name = "Data")]  
public DateTime Date { get; set; }  
public string? UserId { get; set; }  
public IdentityUser? User { get; set; }

### 4. History

Reprezentuje historię aktywności użytkownika.

- **Atrybuty:**  
public int Id { get; set; }  
public DateTime StartDate { get; set; }  
public DateTime EndDate { get; set; }

```
public List<MealsList> Meals { get; set; } = new List<MealsList>();  
public List<ExercisesList> Exercises { get; set; } = new List<ExercisesList>();  
public int TotalCalories { get; set; }  
public int TotalExercises { get; set; }  
public int CaloriesBalance { get; set; }  
public int TotalBurnedCalories { get; set; }
```

## Wylistowane kontrolery wraz z metodami

Kontrolery zarządzają logiką aplikacji i obsługują żądania użytkowników:

1. **DashboardsController.cs**
  - Index() – Wyświetla bilans kalorii i podsumowanie dnia.
2. **ExercisesListsController.cs**
  - Index() – Lista ćwiczeń użytkownika.
  - Create() – Formularz dodawania nowego ćwiczenia.
  - Edit(int id) – Edytowanie wybranego ćwiczenia.
  - Details(int id) – Wyświetla szczegóły ćwiczenia.
  - Delete(int id) – Usunięcie ćwiczenia.
3. **MealsListsController.cs**
  - Index() – Lista posiłków użytkownika.
  - Create() – Dodanie nowego posiłku.
  - Edit(int id) – Edytowanie szczegółów posiłku.
  - Details(int id) – Szczegóły posiłku.
  - Delete(int id) – Usunięcie posiłku.
4. **HistoriesController.cs**
  - Index() – Wyświetla historię posiłków i ćwiczeń z danego dnia.
5. **HomeController.cs**
  - Index() – Strona główna aplikacji.
  - About() – Strona z informacjami o aplikacji.

## Opis systemu użytkowników

System użytkowników bazuje na **Microsoft Identity**, zapewniając:

- **Rejestrację i logowanie** z walidacją danych użytkownika.
- **Haszowanie haseł** i ich bezpieczne przechowywanie w bazie danych.
- **Autoryzację** – dostęp do danych ograniczony tylko do właściciela (UserId).
- Możliwość zarządzania swoimi danymi, w tym posiłkami i ćwiczeniami, w sposób bezpieczny i intuicyjny.

## **Charakterystyka najciekawszych funkcjonalności**

### **1. Bilans kalorii w czasie rzeczywistym**

- W module Dashboard użytkownik może na bieżąco widzieć swój bilans kalorii – ile spożył, ile spalił i jaka jest różnica.

### **2. Historia aktywności**

- Dzięki modułowi History użytkownik może przejrzeć swoje dane z dowolnego dnia – jakie posiłki spożył i jakie ćwiczenia wykonał.

### **3. Elastyczne zarządzanie danymi**

- Posiłki i ćwiczenia mogą być wielokrotnie przypisywane do różnych dni, co oszczędza czas użytkownika.

### **4. Łatwe wprowadzanie danych**

- Intuicyjne formularze do dodawania, edytowania i usuwania posiłków oraz ćwiczeń, zgodne z zasadami CRUD (Create, Read, Update, Delete).

### **5. Bezpieczna obsługa użytkowników**

- Mechanizmy Identity oraz autoryzacji chronią dane przed dostępem nieautoryzowanych użytkowników.