

POLITECHNIKA ŚLĄSKA W GLIWICACH WYDZIAŁ ELEKTRYCZNY

Instytut Elektrotechniki i Informatyki

**PROJEKT INŻYNIERSKI**

**Temat projektu inżynierskiego w języku polskim**

**Temat projektu inżynierskiego w języku angielskim**

Student: **Roger Paul Skrzypczyk**  
Nr albumu: xxxxxx

Studia:. Stacjonarne, I stopnia  
Kierunek: Informatyka  
Specjalność: Inżynieria elektryczna

Prowadzący: [dr inż. Bożena Wieczorek](http://eksperci.polsl.pl/eksperci/szczegoly.php?scbpos=&eid=1154&)  
Recenzent: (tytuł naukowy) Imię NAZWISKO

Spis treści

[Słownik pojęć 3](#_Toc469514353)

[Wstęp 4](#_Toc469514354)

[Technologie użyte w projekcie 7](#_Toc469514355)

[Sposób liczenia makroskładników 8](#_Toc469514356)

[Zapotrzebowanie kaloryczne. 9](#_Toc469514357)

[Postrzeganie Kalori 11](#_Toc469514358)

[Typologia Sheldona 12](#_Toc469514359)

[Implementacja 13](#_Toc469514360)

[Struktura projektu 13](#_Toc469514361)

[Nawigacja 16](#_Toc469514362)

[Baza danych 18](#_Toc469514363)

[Szyfrowanie danych 19](#_Toc469514364)

[Literatura 20](#_Toc469514365)

Słownik pojęć

Framework - Zbiór narzędzi pozwalających na budowę aplikacji i ułatwiający pracę po przez komponenty oraz biblioteki przeznaczone do wykonywania określonych zadań.

Sól (*salt*) – Są to dane losowe dodawane do hasła podczas obliczania [funkcji skrótu](https://pl.wikipedia.org/wiki/Funkcja_skr%C3%B3tu" \o "Funkcja skrótu) . Sól zapopbiega atakowi bazy haseł przy pomocy ataków słownikowych. Jako że jest przechowywana jawnie, nie ma ona znaczenia względem [ataków brute-force](https://pl.wikipedia.org/wiki/Atak_brute_force" \o "Atak brute force).

TDEE – (Total Daily Energy Expenditure) Całkowite dzienne zapotrzebowanie kaloryczne.

BMR – (Basal Metabolic Rate) Wskaźnik podstawowej przemiany materii.

TEA - kalorie spalone przez aktywność fizyczną.

NEAT – Kalorie spalane podczas codziennych czynności.

TEF (Thermic Effect of Food) – Efekt termiczny pożywienia.

Ektomorfik – Typ budowy ciała cechujący się smukłą sylwetką, drobnymi koścmi, chudymi oraz długimi kończynami, a także wąskimi ramionami.

Endomorfik - Typ budowy ciała cechujący się masywną, owalną sylwetką, mający tędencje do tycia po przez wolny metabolizm oraz posiadający wysoki poziom tkanki tłuszczowej.

Mezomorfik – Typ budowy ciała cechujący się szerokim rozstawem barków, długimi kończynami oraz wąską talią, co wizualnie przypomina literę „V”. W łatwy sposób pozyskuje mięśnie oraz posiada szybszą regenerację. Typ ten posiadają nieliczni.

Metabolizm –

Kaloria – W dietetyce kilo kaloria lub „Kaloria” przez duże „K”.Określenie ilość ciepła potrzebną do podgrzania 1 litra wody o 1ºC.

makroskładniki

Wstęp

Na przełomie kilku lat, zmienił się pogląd ludzi dotyczący zdrowego trybu życia. Zauważalne jest ich samozaparcie, dyscyplina w dążeniu do wymażonej sylwetki oraz długiego życia. Przyczyniło się to do większego zapotrzebowanie na strony przeznaczone tej dziedzinie. Jednym z głównych czynników zdrowego trybu życia jest dieta. Często błędnie postrzegana jako deficyt kaloryczny. Dieta może służyć zarówno utracie jak i przyroście masy ciała. Wyróżniamy kilka sposobów kontroli diety. Możliwe jest zlecenie dietetykowki sporządzenia rozpiski posiłków lub wyboru gotowych szablonów. Jednym z popularniejszych sposobów w ostatnich latach jest liczenie kalorii oraz makroskladników. Jest to w zupełności wystarczające dla osób które po prostu chcą zwiększyć lub zmniejszyć masę ciała. Istnieje wiele stron poświęconych liczeniu kalorii. Większość posiada nadmiar informacji co przyczynia się do zniechęcania użytkowników.

Statystyki dotyczące wagi ludności w Polsce przeprowadzone na przełomie 10 lat:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Województwo | Nadwaga i otyłość | Niedowaga | Prawidłowa waga | Nadwaga | Otyłość | Trudno powiedzieć |
| Sląskie | 44,31 | 3,53 | 44,67 | 32,48 | 11,83 | 0,08 |
| Małopolskie | 41,90 | 3,73 | 48,60 | 33,17 | 8,73 | 5,77 |
| Podkarpackie | 46,04 | 3,29 | 46,52 | 34,20 | 11,84 | 4,15 |
| Lubelskie | 43,10 | 4,74 | 44,47 | 30,51 | 12,59 | 7,68 |
| Mazowieckie | 44,12 | 3,03 | 45,55 | 32,31 | 11,81 | 7,31 |
| świętokrzyskie | 42,38 | 3,61 | 50,58 | 33,22 | 9,16 | 3,44 |
| Opolskie | 47,78 | 2,36 | 42,52 | 33,45 | 14,33 | 7,33 |
| Dolnośląskie | 44,78 | 4,21 | 43,75 | 32,53 | 12,25 | 7,25 |
| Lubuskie | 42,37 | 4,21 | 46,50 | 30,76 | 11,61 | 6,93 |
| Łódzkie | 45,36 | 3,33 | 48,04 | 32,85 | 12,51 | 3,28 |
| Wielkopolskie | 46,29 | 3,41 | 44,60 | 32,51 | 13,78 | 5,70 |
| Kujawsko-Pomorskie | 43,77 | 3,98 | 47,22 | 32,05 | 11,72 | 5,04 |
| Zachodniopomorskie | 43,30 | 3,41 | 44,04 | 30,81 | 12,49 | 9,24 |
| Warmińskoi-Mazurskie | 47,17 | 3,56 | 42,41 | 33,70 | 13,47 | 6,86 |
| Podlaskie | 43,22 | 3,52 | 45,63 | 29,97 | 13,25 | 7,63 |
| Pomorskie | 42,52 | 3,09 | 45,03 | 31,07 | 11,45 | 9,36 |

Tabela 1.0 Statystyki w Polsce na przełomie 07.2006-06.2007

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Województwo | Nadwaga i Otyłość | Niedowaga | Prawidłowa waga | Nadwaga | Otyłość | Trudno powiedzieć |
| Sląskie | 48,14 | 2,47 | 40,45 | 36,96 | 11,18 | 8,94 |
| Małopolskie | 40,96 | 2,6 | 48,96 | 30,88 | 10,08 | 7,47 |
| Podkarpackie | 49,69 | 2,06 | 44,98 | 39,5 | 10,19 | 3,27 |
| Lubelskie | 46,54 | 3,17 | 43,72 | 32,61 | 13,93 | 6,57 |
| Mazowieckie | 48,47 | 2,08 | 44,05 | 35,19 | 13,28 | 5,4 |
| świętokrzyskie | 48,33 | 2,82 | 47,41 | 37 | 11,33 | 1,39 |
| Opolskie | 47,04 | 1,27 | 48,34 | 38,06 | 8,98 | 3,34 |
| Dolnośląskie | 49,74 | 2,36 | 44,44 | 36,01 | 13,73 | 3,47 |
| Lubuskie | 49,51 | 2,86 | 42,5 | 32,82 | 16,69 | 5,13 |
| Łódzkie | 53,22 | 1,33 | 41,33 | 40,39 | 12,83 | 4,13 |
| Wielkopolskie | 49,99 | 2,63 | 42,94 | 37,34 | 12,65 | 4,44 |
| Kujawsko-Pomorskie | 51,04 | 2,5 | 41,01 | 37,58 | 13,46 | 5,44 |
| Zachodniopomorskie | 43,97 | 2,85 | 43,21 | 32,41 | 11,56 | 9,97 |
| Warmińskoi-Mazurskie | 51,08 | 1,25 | 45,24 | 37,27 | 13,81 | 2,43 |
| Podlaskie | 49,66 | 1,86 | 41,25 | 34,97 | 14,69 | 7,23 |
| Pomorskie | 48,19 | 3,05 | 42,86 | 34,97 | 13,22 | 5,91 |

Tabela 1.1 Statystyki w Polsce na przełomie 07.2011-06.2012

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Województwo | Nadwaga + Otyłość | Niedowaga | Prawidłowa waga | Nadwaga | Otyłość | Trudno powiedzieć |
| Sląskie | 50,68 | 1,06 | 45,83 | 42,11 | 8,57 | 2,43 |
| Małopolskie | 47,29 | 2,39 | 45,86 | 37,13 | 10,16 | 4,46 |
| Podkarpackie | 53,32 | 2,23 | 42,20 | 40,41 | 12,91 | 2,25 |
| Lubelskie | 45,53 | 3,61 | 47,20 | 34,44 | 11,09 | 3,66 |
| Mazowieckie | 49,49 | 1,48 | 41,13 | 36,01 | 13,48 | 7,91 |
| świętokrzyskie | 54,77 | 3,34 | 40,45 | 46,66 | 8,11 | 1,44 |
| Opolskie | 34,81 | 2,81 | 46,48 | 21,23 | 13,58 | 15,90 |
| Dolnośląskie | 51,93 | 1,71 | 42,80 | 37,58 | 14,35 | 3,56 |
| Lubuskie | 44,87 | 2,70 | 51,18 | 29,02 | 15,85 | 1,25 |
| Łódzkie | 52,62 | 1,18 | 44,13 | 38,49 | 14,13 | 2,07 |
| Wielkopolskie | 52,95 | 3,92 | 38,72 | 38,47 | 14,48 | 4,40 |
| Kujawsko-Pomorskie | 47,99 | 4,08 | 44,96 | 36,78 | 11,21 | 2,98 |
| Zachodniopomorskie | 44,90 | 0,73 | 44,88 | 33,55 | 11,35 | 9,48 |
| Warmińskoi-Mazurskie | 45,87 | 1,94 | 49,91 | 26,25 | 19,62 | 2,28 |
| Podlaskie | 46,23 | 2,80 | 47,68 | 32,95 | 13,28 | 3,29 |
| Pomorskie | 51,26 | 1,58 | 40,06 | 36,24 | 15,02 | 7,10 |

Tabela 1.2 Statystyki w Polsce na przełomie 07.2015-12.2015

W przeciągu dziesięciu lat w naszym kraju procent niedowagi zmniejszył się. Przyczyniło się to także do znacznego przyrostu osób z nadwagą oraz otyłością. Jest to niekożystne zjawisko występujące w Polsce jak i na całym świecie.

Najgorzej pod tym względem sytuacja przedstawia się w krajach najbardziej rozwiniętych. Liderem w rankingu ilości otyłych osób w kraju są Stany Zjednoczone gdzie aż 1/3 społeczeństwa ma problemy z nadwagą. Przyczyną tego zjawiska jest udoskonalenie świata, skutkuje to brakiem aktywności fizycznej w społeczeństwie.

Technologie użyte w projekcie

* Visual studio 2015
* Github – Odpowiedzialny za kontrolowanie wersji projektu i wszystkich zmian z nim związanych.
* SourceTree – Narzędzie pomocnicze pozwalające na lepszą wizualizację i przyśpiesznie pracy związanej z kontrolowaniem wersji projektu.
* ASP.NET.Core 1.0 – to nowy framework firmy Microsoft, którego głównymi założeniami są:

1. Otwarte oprogramowanie (Open-Source). Jest to bardzo korzystne dla rozwoju projektu. Każdy użytkownik może mieć wpływ na kod źródłowy co przyczyni się do większej niezawodności oraz optymalizacji oprogramowania.
2. Obsługa multiplatformowa dzięki czemu nie tylko użytkownicy Windowsa ale także iOS i Linuxa będą mieli możliwość pracy z frameworkiem.
3. Modularność poprzez paczki NuGeta. Umożliwia wstrzykiwanie do projektu tylko tych modułów, które są potrzebne. Możliwość wyłączenia nawet podstawowych funkcjonalności min. sesji, MVC czy też używanie plików statycznych. Pozwala to na przyśpieszenia pracy jak i kompilacji samego frameworka poprzez usunięcie zbędnego kodu.

* Semantic UI – Narzędzie deweloperskie służące do pomocy w tworzeniu ładnego,responsywnego układu strony.
* Entity framework Core – Lekkie, rozszeżalne i multiplatformowe narzędzie bazujące na Entity Framework. Pozwala na odtworzeniu relacji w bazie danych za pomocą modeli w aplikacji.

Sposób liczenia makroskładników

Makroskładniki powinny zostać dobrane indywidualnie w zależności od celu i zapotrzebowania energetycznego danej osoby. Podstawowymi makroskładnikami diety są białka, węglowodany i tłuszcze.

Zakłada się że na:

* 1 gram białka przypada 4 kcal
* 1 gram węglowodanów przypada 4 kcal
* 1 gram tłuszczów przypada 9 kcal

Dla osób ćwiczących podstawowymi dietami wynikającymi z podziału makroskładników są:

* Dieta wysokowęglowodanowa
* Dieta wysokotłuszczowa

Główną różnicą między tymi dietami jest stosunek między węglowodanami a tłuszczami. Standardowo zakłada się że osoba ćwiczące powinna spożywać 2g białka na kg masy ciała.

Dla pierwszej diety ilość tłuszczów w diecie nie powinna przekraczać 50g natomiast resztę kalorii uzupełniają węglowodany. Dla drugiej diety ilość białka pozostaje bez zmian natomiast stosunek węgli do tłuszczów jest odwrotnie proporcjonalny względem pierwszej.

Dla przykładu diety wysokowęglowodanowej. Osoba ważąca 70 kg i spożywająca 2500 kCal powinna zjeść około 140g białka , 40g tłuszczy i 395g węglowodanów każdego dnia. W przypadku diety wysokotłuszczowej 140g białka, 40g węglowodanów i 198g tłuszczy.

Zapotrzebowanie kaloryczne.

Występuje kilka sposobów na wyliczenie zapotrzebowania kalorycznego.

1. Jest to najprostszy wzór który w wielu przypadkach się nie sprawdza.
   1. Mnożymy wagę przez dwadzieścia cztery godziny przez co otrzymujemy (BMR) wskaźnik przemiany materii.
   2. Wskaźnik BMR mnożymy przez współczynnik aktywności fizycznej (w przedziale od 1.0 do 2.0) i otrzymujemy zapotrzebowanie kaloryczne.

Przykładowej współczynnik aktywności:

1,0 – leżący lub siedzący tryb życia, brak aktywności fizycznej  
1,2 – praca siedząca, aktywność fizyczna na niskim poziomie  
1,4 – praca niefizyczna, trening 2 razy w tygodniu  
1,6 – lekka praca fizyczna, trening 3-4 razy w tygodniu  
1,8 – praca fizyczna, trening 5 razy w tygodniu  
2,0 – ciężka praca fizyczna, codzienny trening

1. Bardziej złożona metoda która jest opisana wzorem:

TDEE = BMR + TEA + NEAT + TEF

BMR – W zależności od płci:

Dla Kobiet  
 BMR = (9,99 x waga (kg)) + (6,25 x wzrost (cm)) – (4,92 x wiek) – 161

Dla Mężczyzn  
 BMR = (9,99 x waga (kg)) + (6,25 x wzrost (cm)) – (4,92 x wiek) + 5

TEA - W zależności od rodzaju treningu:

Trening siłowy - 7–9 kcal na minutę w zależności od intensywności.  
Trening aerobowy - 5–10 kcal na minutę w zależności od intensywności

NEAT – Zależna od typu budowy ciała:

700-900 kcal – ektomorfik,  
400-500 kcal – mezomorfik,  
200-400 kcal – endomorfik.

TEF – 6-10% \* (TDEE)

Przykładowe wyliczenie dla osoby ważącej 70 kg, w wieku 22 lat o typie ciała endomorfik, która wykonuje pracę fizyczną i trenuje 4 razy w tygodniu po 90 minut dla średniej intensywności:

1. BMR = (9,99 x 70 (kg)) + (6,25 x 178 (cm)) – (4,92 x 22) + 5
2. BMR = 699,3 + 1112,5 – 108,24 + 5
3. BMR = 1708,56 kcal
4. TEA = (4 x 90 min x 7) / 7
5. TEA = 360 kcal
6. NEAT = 300 kcal
7. TEF = (BMR + TEA + NEAT) \* 0.1
8. TEF = 1708,56 kcal + 360 kcal + 300 kcal \* 0.1
9. TEF = 2368,56 \* 0.1
10. TEF = 236,85
11. TDEE = BMR + TEA + NEAT + TEF
12. TDEE = 1708,56 kcal + 360 kcal + 300 kcal + 236,85 kcal
13. TDEE = 2605,41
14. W zależności od próby nabrania lub utraty masy ciała należy do końcowego wyniku dodać lub odjąć 300 kalorii.

Postrzeganie Kalori

Istnieje wiele kontrowersji dotyczących pomiaru spożytego jedzenia w postaci kcal. Sama jednostka powstała około 130 lat temu, naukowcy spierają się co do jej wiarygodności.

Dowodami na obalenie teorii kalorii sa:

1. Badania prof. Charlesa Libera wykonane w latach 80 za pomocą alkoholu uważanego za bombę kaloryczną udowodniły,że długotrwałe spożywanie dużych ilości alkoholu nie ma wpływu na masę ciała.
2. Na przełomie 20 lat przeprowadzono badania które wykazały, że orzechy nie przyczyniają się do nadmiernego przyrostu masy ciała, a ich spożywanie może być pomocne w redukcji masy ciała.
3. W 2003 roku przepoprawdzono badania na grupie 50 latków którzy spożywali tą samą pule kalorii, różniącą się od siebie składem makroskładników. Różnica wynikała między rozkładem węglowodanów a tłuszczów. Ku zdumieniu wszystkich okazało się że po 12 tygodniach osoby będące na diecie wysokowęglowodanowej schudły 8 kg, a na tłuszczowej 10 kg.

Sa też zwolenicy min. prof. Thomas Sanders z King's College London, który uważa, że kaloria to kaloria.

Prof. Alan Jackson z Uniwersytetu w Southampton uważa że obliczenia dokładnego zapotrzebowania kalorycznego dla danej osoby jest ciężkie jednak aktualny sposób na obliczanie kalorii sprawdza się i jest w zupełności wystarczający.

Nie należy na ślepo wierzyć kalorią natomiast nie należy też ich skreślać. Połączenie podejmowania rozsądnych decyzji żywieniowych wraz z obliczaniem spożytych kalorii przyczyni się do osiągnięcia zamierzonych celów dotyczących sylwetki.

Typologia Sheldona

**William Herbert Sheldon, Jr –** Urodzony 19 listopada 1898, zmarł 17 września 1977 roku. Był amerykańskim psychologiem oraz numizmatą. Stworzył topologię która zakładała że każdy człowiek należy do jednej z trzech grup budowy ciała.

Typ sylwetki każdego człowieka jest przedstawiony za pomocą trzech cyfr, od 1 do 7 , który określa poziom nasilenia w przypadku pierwszej endomorfii, drugiej mezomorfii i trzeciej ektomorfii.

Trzy skrajne typy budowy ciała człowieka.

* typ ektomorficzny (ektomorfik) – numer 117, jest to osoba wysoka, smukła
* typ mezomorficzny (mezomorfik) – numer 171, jest to osoba postawna, umięśniona
* typ endomorficzny (endomorfik) – numer 711, jest to osoba niska, krępa

Typy te są wysokim natężeniem jednej z cech kosztem pozostałych, większość ludzi należy do typów mieszanych.W teorii istnieje 7³ = 343 możliwych typów budowy ciała człowieka. W praktyce liczba ta zaokrągla się do około 80 typów.

Implementacja

W rodziale tym zostanie przedstawiona implementacja strony internetowej wraz z implementacją architektury.

Struktura projektu



Aplikacja powstała na platformie Asp.net.Core. Architektura projektu opiera się na słynnym wzorcu architektonicznym MVC –Model-Widok-Kontroler.

Dodatkowo projekt został podzielony na moduły co w łatwy sposób pozwolił nad odseparowanie funkcjonalności w dalszym rozwoju projektu, a także w razie potrzeby wymiany całego wyglądu aplikacji po przez przepięcie modułów WebGUI.

Projekt składa się z modułów głownych PersonalTrainerCore przechowujące kontrolery oraz PersonalTrainerCore.WebGUI zarządzający widokiem. Posiada także moduły diety PersonalTrainerDiet oraz PersonalTrainerDiet.WebGUI.

Podział na moduły został zaimplementowany przy pomocy klasy pomocniczej przechowująca informację dotyczące danego modułu.

internal class ModuleInfo

{

public Assembly Assembly { get; set; }

public string Name { get; set; }

public string Path { get; set; }

}

Metody GetModules pozwalająca na wydobycie wszystkich modułów ze struktury projektu.

private List<ModuleInfo> GetModules(IHostingEnvironment hostingEnviroment)

{

List<ModuleInfo> modules = new List<ModuleInfo>();

var moduleRootFolder = new DirectoryInfo(Path.Combine(hostingEnviroment.ContentRootPath,

"Modules"));

var moduleFolders = moduleRootFolder.GetDirectories();

foreach (var moduleFolder in moduleFolders)

{

var binFolder = new DirectoryInfo(Path.Combine(moduleFolder.FullName,"bin"));

if (!binFolder.Exists)

continue;

foreach (var file in binFolder.GetFileSystemInfos(moduleFolder.Name + ".dll",

SearchOption.AllDirectories))

{

Assembly assembly = null;

try

{

assembly = AssemblyLoadContext.Default.

LoadFromAssemblyPath(file.FullName);

catch (FileLoadException ex)

{

if (ex.Message == "Assembly with same name is already loaded")

assembly = Assembly.Load(

new AssemblyName(Path.GetFileNameWithoutExtension(file.Name)));

else

throw;

}

if (assembly.FullName.Contains(moduleFolder.Name))

{

if(!modules.Any(x => x.Name.Equals(moduleFolder.Name)))

modules.Add(new ModuleInfo

{

Name = moduleFolder.Name,

Assembly = assembly,

Path = moduleFolder.FullName

});

}

}

}

return modules;

}

Algorytm wydobycia poszczególnych modułów z solucji:

1.Wyszukanie folderu Modules w strukturze aplikacji w projekcie głównym PersonalTrainer.

2.Pobranie z katalogu Modules wszystkich ścieżek modułów.

3.Iteracja pozyskanych ścieżek w celu wyszukania folderów bin.

4.Wydobycie wszystkich plików dll. Znajdujących się w folderach bin.

5.Odczyt z pliku dll asembly zawierających informacje o modułach.

6.Dodanie do listy informacji o modułach które jeszcze nie zostały dodane.

7. Zwrócenie modułów.

8. Dodanie listy do serwisu Mvc dostarczonego przez framework.

Serwis MVC posiada zaimplementowany mechanizm AddApplicationPart dodający wyszukiwanie plików po dodatkowych plikach assembly. Dzięki temu w łatwy sposób możemy wpiąć lub wypiąć moduł w zależności od konfiguracji startowej aplikacji.

Nawigacja

W aplikacji zastosowana została standardowa nawigacja za pomocą MapRoute występującego w serwisie MVC.

app.UseMvc(routes =>

{

routes.MapRoute(

name: "default",

template: "{controller=Home}/{action=Index}/{id?}");

});

Został zdefiniowany szablon wraz ze startową ścieżką strony Home/Index.

Standardowo w aplikacji ASP.NET.CORE kontrolery i widoki są poszukiwane w obrębie głownego projektu w folderach Controllers oraz Views. Ze względu na podział aplikacji na moduły i odseparowanie widoku zostało zadeklarowane rozszeżenie wyszukujące widoków

Nawigacja w poszukiwaniu widoków została skonfigurowana za pomocą

services.Configure<RazorViewEngineOptions>(options =>

{

options.ViewLocationExpanders.Add(

new ModuleViewLocationExpander());

});

Oraz klasy ModuleViewLocationExpander rozszeżającej wyszukiwanie.

public class ModuleViewLocationExpander : IViewLocationExpander

{

private const String \_moduleKey = "module";

public IEnumerable<String> ExpandViewLocations(ViewLocationExpanderContext context, IEnumerable<String> viewLocations)

{

if (context.Values.ContainsKey(\_moduleKey))

{

var module = context.Values[\_moduleKey];

if (!String.IsNullOrWhiteSpace(module))

{

List<String> moduleViewLocations = new List<String>();

moduleViewLocations.Add("/Modules/" + module + ".WebGUI/Views/{1}/{0}.cshtml");

moduleViewLocations.Add("/Modules/" + module + ".WebGUI/Views/Shared/{0}.cshtml");

if (module != "PersonalTrainerCore")

moduleViewLocations.Add("/Modules/" + "PersonalTrainerCore" + ".WebGUI/Views/Shared/{0}.cshtml");

viewLocations = moduleViewLocations.Concat

(viewLocations);

}

}

return viewLocations;

}

}

Metoda ExpandViewLocations bazuje na wyszukiwanie widoków w obrębie Modułu Core (Rdzennego) jak i modułu w którym aktualnie znajduje się użytkownik. Przeszukuje zarówno foldery Views jak i foldery Views/Shared posiadające widoki współdzielone dla danego modulu a w przypadku Core dla całej aplikacji.

public void PopulateValues(ViewLocationExpanderContext context)

{

var controller = context.ActionContext

.ActionDescriptor

.DisplayName;

var moduleName = controller.Split('.')[0];

if (moduleName != "PersonalTrainer")

context.Values[\_moduleKey] = moduleName;

}

Metoda PopulateValues pozwala na zapisanie w zmiennej kontekstowej aktualnego modułu w którym znajduje się użytkownik, która potem zostaje wykorzystana w celu wyszukania widoku w metodzie ExpandViewLocations.

Baza danych

Połaczenie bazy danych z modelami biznesowymi zostało zrealizowane za pomocą Entity Framework Core. Modele zostają mapowane na poziomie kontekstu. W aplikacji został utworzony pojedynczy kontekst o nazwie DefaultContext.

Kontekst jest definiowany podczas startowej konfiguracji aplikacji.

services.AddDbContext<DefaultContext>(

options => options.UseSqlServer(connection, b => b.MigrationsAssembly("PersonalTrainer")), ServiceLifetime.Scoped);

Relacje bazodanowe są zawarte zarówno za pomocą atrybutów wpinanych na poszczególne modele tabel jak i fluent appi z entity frameworku wykorzystywane na poziomie kontekstu.. Należy pamiętać że nie każdą zależność można ustawić za pomocą atrybutów.

Cała logika biznesowa wraz z połaczeniem z bazą danych odbywa się w projekcje Framework.



Szyfrowanie danych

Zapis hasła w bazie danych w momencie rejestracji użytkownika:

1. Wprowadzenie hasła przez użytkownika systemu podczas rejestracji
2. Wygenerowanie soli w celu zaszyfrowania hasła za pomocą wbudowanej klasy RandonNumberGenerator w frameworku .Net

private Byte[] CreateSalt(Int32 size)

{

RandomNumberGenerator rng = RandomNumberGenerator.Create();

Byte[] buff = new Byte[size];

rng.GetBytes(buff);

return buff;

}

1. Wykorzystane funkcji hashująceh **SHA512** na haśle i dodaniem do niego soli.

private Byte[] GenerateSaltedHash(Byte[] plainText, Byte[] salt)

{

HashAlgorithm algorithm = SHA512.Create();

Byte[] plainTextWithSaltBytes =

new Byte[plainText.Length + salt.Length];

for (Int32 i = 0; i < plainText.Length; i++)

plainTextWithSaltBytes[i] = plainText[i];

for (Int32 i = 0; i < salt.Length; i++)

plainTextWithSaltBytes[plainText.Length + i] = salt[i];

return algorithm.ComputeHash(plainTextWithSaltBytes);

}

1. Zapis hashu oraz soli w bazie danych.

Walidacja hasła podanego przez użytkownika.

1. Pobranie z bazy danych hashu oraz soli dla podanego użytkownika.
2. Wygenerowanie hashu za pomocą funkcji hashującej przy pomocy pobranej soli oraz hasła podanego przez użytkownika.
3. Porównanie hashu przechowywanego w bazie danych z wygenerowanym w czasie próby logowania przez użytkownika.

Literatura

<http://oczymlekarze.pl/zdrowy-styl-zycia/dieta/1417-kaloria-kalorii-nierowna>

<http://www.sfd.pl/Ektomorfik,_Endomorfik_i_Mezomorfik_Sylwetka_prawd%C4%99_Ci_powie_-t236043.html>

<https://pl.wikipedia.org/wiki/Typologia_Sheldona>

<http://antropologia-fizyczna.pl/antropometria/typy-budowy-ciala/somatotypy-wg-sheldona>

http://www.sfd.pl/%5Bart%5D\_TYPOLOGIA\_SHELDONA\_co%C5%9B\_wi%C4%99cej\_o\_typie\_budowy\_cia%C5%82a\_-t635046.html

http://www.fabrykasily.pl/porady-trenerow/zapotrzebowanie-kaloryczne-wyliczenie