

# Aplikacja do Nauki Języków Obcych

Autorzy : Dawid Bazylewicz , Sara Fijołek , Miron Kwiatkowski

4.06.2024

## Spis treści

## Contents

1	Opis funkcjonalny systemu	1
2	Opis technologiczny	2
3	Wyszczególnione wdrożone zagadnienia kwalifikacyjne	4
4	Instrukcja lokalnego i zdalnego uruchomienia systemu	4
5	Wnioski projektowe	5

## 1 Opis funkcjonalny systemu

Aplikacja do nauki języków obcych została zaprojektowana z myślą o użytkownikach chcących efektywnie i interaktywnie uczyć się nowych języków. System umożliwia:

- Rejestracje i logowanie użytkowników.
- Tworzenie i zarządzanie kontem użytkownika.
- Dostęp do różnorodnych lekcji i ćwiczeń językowych.
- Śledzenie postępów w nauce.
- Interakcje z innymi użytkownikami poprzez fora dyskusyjne.
- Otrzymywanie powiadomień mailowych dotyczących postępów i nowości.

## 2 Opis technologiczny

System został zbudowany w oparciu o następujące technologie:

### Framework MVC

Wzorzec projektowy Model-View-Controller (MVC) dzieli aplikację na trzy główne warstwy:

- Model - odpowiada za logikę biznesową i dane aplikacji.
- Widok - zajmuje się prezentacją danych i interakcją z użytkownikiem.
- Kontroler - pośredniczy pomiędzy Modelem a Widokiem, przetwarzając żądania i zwracając odpowiednie odpowiedzi.

Projekt wykorzystuje ASP.NET MVC jako framework do realizacji wzorca MVC.

### Framework CSS

Do budowy frontendu zastosowano bibliotekę Bootstrap, która ułatwia tworzenie responsywnych i estetycznych interfejsów użytkownika dzięki gotowym komponentom CSS.

### Baza danych

Dane przechowywane są w Microsoft SQL Server Developer Edition. Do zarządzania bazą danych i przeglądania danych używane jest narzędzie SQL Server Management Studio.

### Cache

System wykorzystuje pamięć podręczną przeglądarki oraz Distributed Memory Cache powiązany z sesją użytkownika.

### Dependency manager

NuGet jest narzędziem zarządzającym zależnościami w projekcie, zapewniając, że wersje bibliotek używanych w środowisku deweloperskim będą zgodne z wersjami w środowisku produkcyjnym. Plik `WebApplication1.csproj` zawiera listę zależności.

## **HTML, CSS, JavaScript**

Do budowy interfejsu użytkownika zastosowano:

- HTML - do strukturyzacji treści.
- CSS - do stylizacji i układu.
- JavaScript - do dynamicznego i interaktywnego działania aplikacji.

## **Routing**

Routing definiuje, jakie akcje mają zostać uruchomione w odpowiedzi na określone żądania URL.

## **ORM**

System korzysta z Object Relational Mapping (ORM), co umożliwia mapowanie obiektów aplikacji na tabele w bazie danych. Klasy Models odpowiadają tabelom w bazie danych, a DbContext zarządza kontekstem danych.

## **Uwierzytelnianie**

Po zalogowaniu się użytkownika, serwer tworzy sesję i zwraca jej identyfikator, najczęściej poprzez ciasteczka. Sesja użytkownika przechowuje informacje takie jak Email oraz Name.

## **Mailing**

Aplikacja umożliwia wysyłanie powiadomień mailowych do użytkowników.

## **Formularze**

Formularze HTML są wykorzystywane do przesyłania danych użytkownika.

## **Asynchroniczne interakcje**

Asynchroniczne żądania HTTP (AJAX) umożliwiają komunikację z serwerem bez przeładowywania strony, co zapewnia bardziej responsywny interfejs użytkownika.

## **RWD**

Aplikacja została zaprojektowana zgodnie z zasadami Responsive Web Design, dzięki czemu wyświetla się prawidłowo na różnych urządzeniach i rozmiarach ekranów.

## **Logger**

System loguje wszystkie akcje, zapisując informacje, ostrzeżenia, błędy oraz krytyczne wiadomości w konsoli.

## **Deployment**

Aplikacja została wdrożona, umożliwiając jej uruchomienie na różnych systemach.

## **3 Wyszczególnione wdrożone zagadnienia kwalifikacyjne**

- Implementacja wzorca MVC.
- Użycie frameworku Bootstrap do budowy frontendu.
- Konfiguracja i zarządzanie bazą danych Microsoft SQL Server.
- Wykorzystanie mechanizmu Cache do poprawy wydajności.
- Zarządzanie zależnościami przy pomocy NuGet.
- Asynchroniczne interakcje użytkownika z serwerem.
- Responsywność aplikacji dzięki RWD.
- Logowanie działań systemu.

## **4 Instrukcja lokalnego i zdalnego uruchomienia systemu**

### **Lokalne uruchomienie**

1. Zainstaluj środowisko .NET Core SDK oraz SQL Server Developer Edition.

2. Sklonuj repozytorium projektu z systemu kontroli wersji (np. Git).
3. Otwórz projekt w Visual Studio.
4. Skonfiguruj połączenie z baza danych w pliku konfiguracyjnym.
5. Wykonaj migracje bazy danych.
6. Uruchom projekt lokalnie przy użyciu Visual Studio.

## **Zdalne uruchomienie**

1. Skonfiguruj serwer z systemem operacyjnym zgodnym z wymaganiami .NET Core.
2. Zainstaluj SQL Server na serwerze lub skonfiguruj dostęp do zdalnej bazy danych.
3. Sklonuj repozytorium projektu na serwer.
4. Skonfiguruj zmienne środowiskowe dla połączeń z baza danych.
5. Wykonaj migracje bazy danych.
6. Uruchom aplikacje przy użyciu narzędzia do zarządzania procesami (np. IIS, Nginx).

## **5 Wnioski projektowe**

Projektowanie i implementacja aplikacji do nauki języków obcych w oparciu o ASP.NET MVC pozwoliło na stworzenie skalowalnej i wydajnej platformy edukacyjnej. Wykorzystanie współczesnych technologii frontendowych, takich jak Bootstrap i JavaScript, oraz narzędzi zarządzania danymi, takich jak Microsoft SQL Server, umożliwiło zbudowanie intuicyjnego i funkcjonalnego interfejsu użytkownika. Zastosowanie wzorca projektowego MVC zapewniło klarowną strukturę kodu, ułatwiającą rozwój i utrzymanie aplikacji. Wdrożenie mechanizmów logowania oraz zarządzania sesjami użytkowników zapewniło bezpieczeństwo i niezawodność systemu.