Szkoła Sztuk Walki

Projekt przygotowany przez Igora Lelenia, s15833  
TIN Internetowe, semestr letni 2023

# Wymagana zawartość pracy projektowej

* Warstwa prezentacyjna stosująca do określenia wyglądu dokumentów zewnętrzne arkusze CSS  
  - arkusze CSS zostały dodane do folderu **MartialArtsSchool\MartialArtsSchool\wwwroot\css**
* Funkcjonalność obejmująca przechowywanie trwałych danych na serwerze za pomocą wybranego systemu relacyjnych lub nierelacyjnych baz danych. Wskazane jest, aby obsługiwane były dane reprezentowane przez przynajmniej dwie powiązane ze sobą encje (tj. np. Pracownik-Dział, Produkt-Zamówienie itp.)  
  - rozwiązanie
* Przykłady walidacji wprowadzanych przez użytkownika danych –zarówno po stronie klienta jak i serwera  
  - walidacja po stronie klienta wprowadzona w postaci ograniczeń pól w modelach dostępnych w folderze **MartialArtsSchool\MartialArtsSchool\Models**   
  - walidacja po stronie serwera baz danych w postaci triggera oczekującego na otrzymanie pustego Id użytkownika, tak aby dodać go do bazy nadając mu automatyczne Id. Bez tego triggera, warstwa kliencka nadawała automatyczne id, które nie było widoczne dla systemu baz danych  
  - walidacja po stronie serwera wynika także z zależności między modelem baz danych, który został przekonwertowany za pomocą mapowania obiektowo-relacyjnego do warstwy aplikacji webowej. Przy próbie pominięcia walidacji klienckiej poprzez wysłanie żądania o np. zbyt długim nazwisku, system zwróci błąd
* Funkcjonalność wykorzystująca wybraną technikę podtrzymania informacji o stanie współdziałania z klientem (np. za pomocą zmiennych sesyjnych lub cookies)  
  - wprowadzenie do widoku startowego funkcji javascript zapewnienie utrzymanie wyskakującego okienka do logowania jeśli użytkownik zalogował się wcześniej, a dane trzymane są w ciasteczkach. W skrócie: nowy użytkownik otrzyma stronę startową z przyciskiem do logowania, a użytkownik, który wcześniej logował się do systemu zostanie od razu przekierowany do okna logowania
* Elementy funkcjonalności oparte na skryptach działających po stronie przeglądarki klienta (JavaScript –najlepiej przy wsparciu wybranej biblioteki czy ramy JS)  
  - w pliku **MartialArtsSchool\MartialArtsSchool\Shared/\_Layout.cshtml**dodane zostały skrypty javascript służące do walidacji formularzy. Np. skrypt <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validation-unobtrusive/3.2.11/jquery.validate.unobtrusive.min.js"></script>  
  służący do walidacji dyskretnej formularzy HTML-5
* Implementacja i konfiguracja witryny respektuje najważniejsze zalecenia i dobre praktyki związane z bezpieczeństwem aplikacji WWW.  
  - mam szczerą nadzieję

# Opis utworzonej funkcjonalności

Wybrane technologie:

* **Microsoft SQL Server** - platforma bazodanowa pozwalająca na pracę użytkowników z bazą danych w architekturze klient-serwer. Komponenty działające niezależnie wobec siebie zapewniają bezpieczeństwo danych, tworzenie kopii zapasowych, analizowanie i tworzenie raportów. Komunikacja użytkownika z silnikiem bazy jest możliwa za pomocą dedykowanego tej platformie rozszerzenia języka SQL, czyli Transact-SQL.  
  Za pomocą Transact-SQL można również sterować pracą serwera baz danych.
* **ASP .NET Core** - platforma typu open-source rozwijana przez firmę Microsoft.  
  Z założenia ma ona wspierać wieloplatformowość aplikacji, niezależnie od posiadanej przez klienta wersji frameworka. Dzięki temu aplikacja webowa powinna uruchamiać się zarówno na komputerach z systemem Windows, Mac czy Linux.
* **Entity Framework** - narzędzie mapowania obiektowo-relacyjnego. Generuje obiekty biznesowe oraz encje zgodnie z tabelami baz danych. Obiekt biznesowy jest encją w wielowarstwowej aplikacji, która działa w połączeniu z dostępem do bazy danych oraz warstwą logiki biznesowej służącą do przesyłania danych. Z kolei encja odnosi się do czegoś, co jest unikatowe i istnieje oddzielnie, czyli do tabeli bazy danych.
* **Razor Pages** – rozwiązanie powiązane ze środowiskiem ASP .NET Core – umożliwia tworzenie dynamicznych widoków (stron internetowych) wykorzystujących bezpośrednio strukturę obiektów wynikającą z modelu danych. Dzięki temu, można z niej bezpośrednio tworzyć widoki szczegółowe, widok listy oraz widok dla wszystkich operacji CRUD dla obiektów rozwiązania. Strony te później zostały przeze mnie edytowane:
  + Obsługują nowe metody z kontrolerów
  + Ich wygląd się różni
  + Zostały uzupełnione treścią
  + Nadane im zostały dodatkowe walidacje
  + Nadałem im style za pomocą dodatkowych arkuszy CSS

Rozwiązanie wspiera zarządzanie Szkołą Sztuk Walki. Można za jego pomocą dodać nowych członków klubu, nowe kursy, a także edytować obecne oraz podejrzeć ich detale. Niestety, podczas próby stworzenia encji asocjacyjnej, otrzymywałem dziwne błędy serwera, dlatego usunąłem widok odpowiedzialny za zapisywanie uczestników na kurs.

# Wskazówki dotyczące użytkowania

1. Użytkownik zostaje przekierowany na stronę startową na której może się zalogować lub nawigować do stron wyświetlających listy obiektów: Members oraz Lessons:  
   Obraz zawierający tekst, oprogramowanie, Strona internetowa, Czcionka

   Opis wygenerowany automatycznie  
     
   Na stronach obiektów widnieją dane rekordów pobrane z bazy danych, a także akcje pozwalające na wyświetlenie szczegółów rekordów, edycję lub usunięcie. W każdym widoku, u góry, po lewej stronie, znajduje się przycisk umożliwiający utworzenie nowego rekordu.  
   Obraz zawierający tekst, Czcionka, numer, linia

   Opis wygenerowany automatycznie  
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
     
   W stronach szczegółowych do edycji oraz usunięcia, znajduje się opcja potwierdzenia akcji lub powrotu do listy obiektów. Powrót do listy obiektów jest także dostępny w stronie szczegółowej rekordu- tak więc do list można wrócić tym przyciskiem, standardowym przyciskiem wstecz, lub paskiem nawigacyjnym znajdującym się zawsze u góry strony.  
   Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

   Opis wygenerowany automatycznie

# Informacje nt. instalacji i wymagań środowiska niezbędnych dla uruchomienia aplikacji

1. W**ymagania sprzętowe:**

* Pamięć RAM:  
  Minimum 2 GB
* Procesor:  
  Dwurdzeniowy procesor z zegarem 1.9 GHz w architekturze x86 lub x64 z obsługą zestawu instrukcji SSE2 lub lepszy
* Wyświetlacz:  
  Super VGA o rozdzielczości 1024 x 768 lub lepszy
* Łącze internetowe:  
  Przepustowość: większa niż 50 Kb/s (499 Kb/s)  
  Opóźnienia: poniżej 150 ms
* Obsługiwane przeglądarki:
* Google Chrome
* Mozilla Firefox
* Apple Safari

1. Rozwiązanie (cały folder) należy uruchomić używając programu Visual Studio 2022 (w wersjach starszych, np. 2019, nie są obsługiwane niektóre pakiety, jak np. Entity Framework użyte w tym rozwiązaniu). Następnie należy połączyć się z bazą danych klikając dwukrotnie w nazwę połączenia widniejące po lewej stronie, w eksploratorze serwera:  
   Obraz zawierający Oprogramowanie multimedialne, oprogramowanie, tekst, Oprogramowanie graficzne

   Opis wygenerowany automatycznie  
   Kolejnym krokiem jest zbudowanie oraz uruchomienia aplikacji na localhost za za pomocą przycisku na środku, pod panelem nawigacyjnymi programu Visual Studio 2022:  
   Obraz zawierający Oprogramowanie multimedialne, Oprogramowanie graficzne, oprogramowanie, zrzut ekranu

   Opis wygenerowany automatycznie  
     
   **Dopisek wynikający z niemożliwości wrzucenia całego rozwiązania do programu gakko:**

Przesłałem same pliki źródłowe, jednak bez nich wymagana jest instalacja pakietów nuget:  
Obraz zawierający tekst, elektronika, zrzut ekranu, wyświetlacz

Opis wygenerowany automatycznie

Z folderu rozwiązania został usunięty folder BIN, aby zmieścić się w 25MB. tak więc możliwe, że należy stworzyć całe rozwiązanie, a następnie dodać pliki według struktury poniżej:  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie  
  
Ponadto, nie wiem w jaki sposób przekazać serwer bazodanowy (sam schemat, skrypty do tworzenia bazy danych?). Poprosiłbym o komentarz i umożliwienie mi przesłania plików w wymagany sposób.

Podejrzewam, że błąd wynika ze zbyt dużego rozmiaru rozwiązania, dlatego zamieszczam otrzymany przeze mnie błąd (HTTP ERROR 400):  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie