

MoveMentor

1.Opis Działania Projektu

MoveMentor jest aplikacją webową, która umożliwia użytkownikom planowanie aktywności sportowych poprzez dodawanie i dostosowywanie aktywności fizycznych do określonych dat i godzin. Aplikacja ma na celu ułatwienie organizacji harmonogramu treningowego. Możliwe będzie dodatkowo ustawianie odpowiednich do wydarzeń komentarzy jak np. „Siatkówka ze znajomymi”, „Gra w tenisa z Prezydentem” . Istnieje też funkcja dodawania kontaktów np. właśnie do prezydenta, dzięki czemu ponownie można umówić się na wspólny trening. Użytkownicy będą mieć możliwość tworzenia oraz usuwania kont, jak także logowania/wylogowywania się z tych już istniejących. Cały harmonogram, tak jak i wprowadzone aktywności będą zapisywać się do danego konta.

2. Autorzy

Wiktor Nejman

Jakub Stachowicz

3.Specyfikacja wykorzystanych technologii

Wersja .NET: .NET 6.0

Baza danych: Entity Framework Core.

4.Instrukcje Pierwszego Uruchomienia Projektu

Uruchomienie pliku „MoveMentor.sn1”, w przypadku używania VisualStudio należy wpisać w konsoli menadżera pakietów następujące linijki „Add-Migration initial” oraz „Update-Database”.

5.Opis Struktury Projektu

Areas: Zawiera strony związane z tożsamością.

Controllers: Zawiera kontrolery MVC obsługujące żądania HTTP.

Data: Zawiera DbContext i migracje.

Models: Zawiera modele danych.

Properties: Zawiera właściwości i ustawienia projektu.

Views: Zawiera widoki

Razor dla frameworka MVC.

wwwroot: Zawiera pliki statyczne, takie jak CSS, JS i obrazy.

6. Modele

User (Użytkownik)

Opis: Reprezentuje użytkownika aplikacji.

Pola:

Id (int): Klucz główny.

Username (string): Unikalny identyfikator użytkownika.

PasswordHash (string): Zahasowane hasło do uwierzytelniania.

Role (string): Rola użytkownika w systemie (np. Admin, User).

SportType (Ruch/Aktywność)

Opis: Reprezentuje ruch fizyczny lub ćwiczenie.

Pola:

Id (int): Klucz główny.

Name (string): Nazwa ruchu/dyscypliny/aktywności.

Trening

Pola: **Id** (int): Klucz główny.

Name (string): Nazwa treningu.

Comment (string): opis treningu / komentarz do niego.

Start&End Date: Daty i godziny rozpoczęcia i zakończenia aktywności.

Contacts

Pola: **Id** (int): Klucz główny.

Name (string): Nazwa treningu.

Phone (int): Numer telefonu.

Mail (string): Adres e-mail.

7.Kontrolery

UserController (Kontroler Użytkowników)

Metody: GET /users:

Opis: Pobiera listę wszystkich użytkowników.

Parametry: Brak

Zwraca: Lista użytkowników w formacie JSON.

POST /users:

Opis: Tworzy nowego użytkownika.

Parametry: JSON z danymi użytkownika.

Zwraca: Utworzonego użytkownika.

MovementController (Kontroler Ruchów)

Metody: GET /movements:

Opis: Pobiera listę wszystkich ruchów.

Parametry: Brak

Zwraca: Lista ruchów w formacie JSON.

POST /movements:

Opis: Tworzy nowy ruch.

Parametry: JSON z danymi ruchu.

Zwraca: Utworzony ruch.

WorkoutController (Kontroler Treningów)

Metody: GET /workouts:

Opis: Pobiera listę wszystkich treningów.

Parametry: Brak

Zwraca: Lista treningów w formacie JSON.

POST /workouts:

Opis: Tworzy nowy trening.

Parametry: JSON z danymi treningu.

Zwraca: Utworzony trening.

ContactController (Kontroler kontaktów)

Metody: GET /workouts:

Opis: Pobiera listę wszystkich kontaktów.

Parametry: Brak

Zwraca: Lista kontaktów w formacie JSON.

POST /workouts:

Opis: Tworzy nowy kontakt.

Parametry: JSON z danymi kontaktu.

Zwraca: Utworzony kontakt.

8.System użytkowników

Role: Admin, User

Przydzielanie ról: Role mogą być przypisane podczas tworzenia użytkownika lub zaktualizowane przez administratora.

Uprawnienia:

Zalogowani użytkownicy: Mogą przeglądać i tworzyć ruchy, treningi oraz nowe kontakty.

Goście: Mogą tylko przeglądać ruchy.

Informacje powiązane z użytkownikiem: Dane osobiste, takie jak nazwa użytkownika, rola oraz utworzone ruchy/treningi.

Informacje globalne: Publiczne ruchy dostępne dla wszystkich użytkowników.

9. Najciekawsze funkcjonalności

Tworzenie niestandardowych treningów: Użytkownicy mogą tworzyć spersonalizowane rutyny treningowe, wybierając z różnych ruchów, wraz z odpowiednimi komentarzami.

Kontrola dostępu na podstawie ról: użytkownicy zalogowani mają możliwość zarządzania swoimi treningami i kontaktami, a goście jedynie mogą sprawdzać dostępne ruchy, co zapewnia bezpieczny i zorganizowany system zarządzania użytkownikami.

Wszystkie pola mają ustawione ograniczniki, aby zapobiec nieprawidłowościom, dodatkowo data zakończenia treningu nie może być wcześniejsza niż jego początkowa data, a utworzenie treningu nie może odbywać się w przeszłości (5.07 nie utworzymy treningu który ma się odbyć np. 2.07).