

Dokumentacja Techniczna Projektu

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”

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: Prawdopodobnie Entity Framework Core (szczegóły o konkretnej bazie danych, np. SQL Server, SQLite, nie są bezpośrednio widoczne).

4.Instrukcje Pierwszego Uruchomienia Projektu

Przejsie do Katalogu Projektu: cd projekt/MoveMentor

Przywrócenie Pakietów NuGet: dotnet restore

Budowanie Projektu: dotnet build

Aktualizacja Bazy Danych: uruchomić dotnet ef database update

Uruchomienie Projektu: dotnet run

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): Zahasłowane hasło do uwierzytelniania.

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

Movement (Ruch)

Opis: Reprezentuje ruch fizyczny lub ćwiczenie.

Pola:

Id (int): Klucz główny.

Name (string): Nazwa ruchu.

Description (string): Szczegółowy opis ruchu.

Category (string): Kategoria ruchu (np. Cardio, Siłowe).

Difficulty (int): Poziom trudności od 1 do 5.

Workout (Trening)

Opis: Reprezentuje rutynę treningową.

Pola:

Id (int): Klucz główny.

Name (string): Nazwa treningu.

Description (string): Szczegółowy opis treningu.

Movements (List<Movement>): Lista ruchów zawartych w treningu.

Duration (int): Czas trwania w minutach.

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: Utworzony użytkownik.

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.

8.System uzytkownikow

Role: Admin, User

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

Uprawnienia:

Zalogowani użytkownicy: Mogą przeglądać i tworzyć ruchy i treningi.

Goście: Mogą tylko przeglądać ruchy i treningi.

Informacje powiązane z użytkownikiem:

Dane osobiste, takie jak nazwa użytkownika, rola oraz utworzone ruchy/treningi.

Informacje globalne:

Publiczne ruchy i treningi dostępne dla wszystkich użytkowników.

9.Najciekawsze funkcjonalnosci

Najciekawsze funkcjonalności

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

Poziomy trudności: Ruchy są kategoryzowane według trudności, co pomaga użytkownikom wybrać ćwiczenia odpowiednie dla ich poziomu sprawności.

Kontrola dostępu na podstawie ról: Administratorzy mają możliwość zarządzania użytkownikami i ich rolami, co zapewnia bezpieczny i zorganizowany system zarządzania użytkownikami.