

DOKUMENTACJA TECHNICZNA PROJEKTU ZALICZENIOWEGO Z PROGRAMOWANIA ZAAWANSOWANEGO

Adam Kurek, Damian Kaszubowski

Tytuł projektu: ArenaZawodów - Aplikacja do Zarządzania Amatorskimi Zawodami Sportowymi

Opis Projektu:

Aplikacja internetowa umożliwiająca organizację i zarządzanie amatorskimi zawodami sportowymi w różnych dyscyplinach. Projekt skierowany jest zarówno do organizatorów takich wydarzeń, jak i uczestników, którzy w danych wydarzeniach chcą wziąć udział.

Główne funkcjonalności:

1. Rejestracja użytkowników i logowanie:

2. Zarządzanie wydarzeniami sportowymi:

- Tworzenie, edytowanie i usuwanie wydarzeń przez gospodarzy.
- Ustawianie wydarzeń jako nieużywane.

3. Obsługa rejestracji uczestników:

- Uczestnicy mogą rejestrować się na wydarzenia indywidualne lub drużynowe.
- System weryfikuje dostępność miejsc i status rejestracji.

4. Zarządzanie drużynami:

- Tworzenie drużyn przez uczestników w wydarzeniach drużynowych.
- Zapraszanie członków do drużyny.
- Usuwanie członków drużyny lub przekazywanie roli kapitana.

5. Wyświetlanie nadchodzących i archiwalnych wydarzeń:

- Filtracja wydarzeń względem dyscypliny.
- Wyświetlanie najważniejszych informacji i możliwość zajrzenia w szczegóły.
- W tym wyświetlanie aktualnego statusu rejestracji.

6. Powiadomienia:

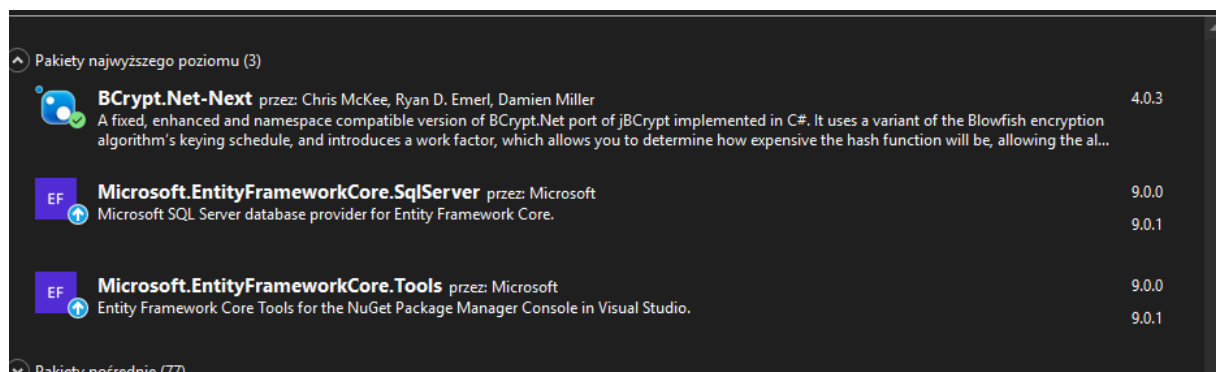
- System powiadomień zliczający nieprzeczytane wiadomości.
- Informacje o zaproszeniach do drużyn i innych działaniach związanych z kontem użytkownika.

Specyfikacja wykorzystanych technologii:

1. Język programowania C#
2. Framework: ASP.NET Core 8 (Model-View-Controller)
3. Środowisko programistyczne: Visual Studio 2022
4. Baza danych: Microsoft SQL Server LocalDB
5. System zarządzania powiadomieniami: JavaScript

Instrukcje pierwszego uruchomienia projektu:

Aby uruchomić projekt, należy otworzyć plik *SportsEventManager.sln* w Visual Studio 2022, pobrać poniższe rozszerzenia NuGet, w Package Manager Console wykonać polecenie *Update-Database*, ustawić projekt jako startowy, nacisnąć F5, a aplikacja uruchomi się pod adresem <https://localhost:xxxx>.



Struktura projektu ArenaZawodów:

1. Controllers - Zawiera kontrolery odpowiadające za logikę aplikacji i obsługę żądań HTTP.
2. Models - Zawiera klasy reprezentujące modele bazy danych w aplikacji.
3. ViewModels - Zawiera modele używane do przekazywania danych między kontrolerami a widokami.
4. View - Folder z widokami w technologii Razor, podzielony na podfoldery odpowiadające kontrolerom. Każdy widok renderuje interfejs użytkownika dla akcji kontrolera.

5. Data – Zawiera ApplicationDbContext, który jest kontekstem bazy danych, definiującym połączenia z MS SQL oraz relacje między modelami.

6. Pliki konfiguracyjne – konfiguruje aplikację i jej zależności.

7. Dodatkowo niektóre pliki zawierają zasoby statyczne takie jak CSS i JavaScript.

Lista modeli i ich opis:

1. Event: Reprezentuje wydarzenie sportowe, jego dane podstawowe, terminy, lokalizację i ograniczenia uczestników.

- **Id**: Identyfikator wydarzenia.
- **Name**: Nazwa wydarzenia (wymagane, maks. 100 znaków).
- **Date**: Data i godzina wydarzenia (wymagana, musi być w przyszłości).
- **Location**: Lokalizacja wydarzenia (wymagane, maks. 200 znaków).
- **Description**: Szczegółowy opis wydarzenia (wymagane, maks. 10 000 znaków).
- **Capacity**: Maksymalna liczba uczestników (wymagana, zakres 1-1000).
- **OccupiedSeats**: Liczba zajętych miejsc (domyślnie 0).
- **RegistrationStart**: Data rozpoczęcia rejestracji (wymagana, wcześniejsza niż zakończenie).
- **RegistrationEnd**: Data zakończenia rejestracji (wymagana, pomiędzy rozpoczęciem a datą wydarzenia).
- **ParticipationType**: Typ uczestnictwa (Individual lub Team, wymagane).
- **TeamSize**: Liczba osób w drużynie (opcjonalne, zakres 1-30).
- **Discipline**: Dyscyplina wydarzenia (wymagane, maks. 100 znaków).
- **IsArchived**: Czy wydarzenie jest zarchiwizowane (domyślnie false).
- **IsInactive**: Czy wydarzenie jest nieużywane (domyślnie false).

Relacje:

- **EventRegistrations**: Lista rejestracji na wydarzenie.
- **Teams**: Lista drużyn przypisanych do wydarzenia.
- **HostId**: Identyfikator gospodarza.
- **Host**: Relacja do użytkownika-gospodarza.

Metody:

- **RegistrationStatus:** Status rejestracji (start, koniec, zamknięta).
- **CompetitionType:** Typ zawodów (Indywidualne lub Drużynowe).

2. EventRegistration: Rejestracja uczestnika lub drużyny na wydarzenie.

- **Id:** Identyfikator rejestracji.
- **EventId:** Identyfikator wydarzenia.
- **Event:** Relacja do wydarzenia.
- **UserId:** Identyfikator użytkownika (opcjonalne dla drużynowych).
- **User:** Relacja do użytkownika.
- **TeamId:** Identyfikator drużyny (opcjonalne dla indywidualnych).
- **TeamName:** Nazwa drużyny (opcjonalne).
- **RegistrationDate:** Data rejestracji (domyślnie obecna).

3. Notification: Informacje powiadamiające użytkowników o zdarzeniach w systemie.

- **Id:** Identyfikator powiadomienia.
- **UserId:** Identyfikator użytkownika otrzymującego powiadomienie.
- **User:** Relacja do użytkownika.
- **Message:** Treść powiadomienia.
- **Link:** Odnośnik związany z powiadomieniem (opcjonalne).
- **IsRead:** Status przeczytania powiadomienia (domyślnie false).
- **CreatedAt:** Data utworzenia powiadomienia.
- **NotificationDate:** Data powiązanego zdarzenia.

4. Team: Reprezentuje drużynę zarejestrowaną na wydarzenie.

- **Id:** Identyfikator drużyny.
- **TeamName:** Nazwa drużyny.
- **Capacity:** Maksymalna liczba członków.

- **CaptainId**: Identyfikator kapitana drużyny.
- **EventId**: Identyfikator przypisanego wydarzenia.

Relacje:

- **Event**: Relacja do wydarzenia.
- **TeamMembers**: Lista członków drużyny.

5. TeamInvitation: Zaproszenie do drużyny.

- **Id**: Identyfikator zaproszenia.
- **TeamId**: Identyfikator drużyny zapraszającej.
- **Team**: Relacja do drużyny.
- **InvitedUserId**: Identyfikator zaproszonego użytkownika.
- **InvitedUser**: Relacja do użytkownika.
- **Status**: Status zaproszenia (Pending, Accepted, Rejected).

6. TeamMember: Członek drużyny.

- **Id**: Identyfikator członka.
- **TeamId**: Identyfikator drużyny.
- **Team**: Relacja do drużyny.
- **IsCaptain**: Czy użytkownik jest kapitanem (domyślnie false).
- **UserId**: Identyfikator użytkownika.
- **User**: Relacja do użytkownika.

7. User: Reprezentuje użytkownika aplikacji.

- **Id**: Identyfikator użytkownika.
- **FirstName**: Imię użytkownika (wymagane, maks. 50 znaków).
- **LastName**: Nazwisko użytkownika (wymagane, maks. 50 znaków).
- **Email**: Adres e-mail (wymagany, poprawny format).
- **PasswordHash**: Hasło użytkownika (wymagane, min. 6 znaków).
- **PhoneNumber**: Numer telefonu (wymagany, poprawny format).
- **AccountType**: Typ konta (User lub Host, wymagane).

Relacje:

- **TeamMembers:** Lista drużyn, do których należy użytkownik.
- **Invitations:** Lista zaproszeń do drużyn.

8. PastTeams: Model przechowujący informacje o historii członkostwa użytkowników w drużynach podczas różnych wydarzeń.

- **Id:** Unikalny identyfikator rekordu.
- **UserId:** Identyfikator użytkownika, który należał do drużyny. Powiązany z modelem User.
- **TeamId:** Identyfikator drużyny, w której użytkownik był członkiem. Powiązany z modelem Team.
- **EventId:** Identyfikator wydarzenia, w którym drużyna brała udział. Powiązany z modelem Event.
- **JoinDate:** Data dołączenia użytkownika do drużyny.
- **LeaveDate:** Data opuszczenia drużyny przez użytkownika (opcjonalne).

Opis metod w Kontrolerach:

1.AccountController:

Register (GET)

Parametry: Brak

Opis: Wyświetla widok formularza rejestracji.

Zwracane dane: Widok RegisterViewModel.

Register (POST)

Parametry: RegisterViewModel model (Model danych zawierający dane rejestracyjne użytkownika).

Opis: Tworzy nowego użytkownika na podstawie przesłanych danych. Sprawdza, czy adres e-mail już istnieje.

Zwracane dane: Sukces: Przekierowanie do widoku logowania. Błąd: Powrót do widoku z błędami walidacyjnymi.

Login (GET)

Parametry: Brak

Opis: Wyświetla widok logowania.

Zwracane dane: Widok LoginViewModel.

Login (POST)

HTTP: POST

Parametry: LoginViewModel model (Model danych zawierający dane logowania użytkownika).

Opis: Weryfikuje dane logowania, ustawia sesję użytkownika i loguje go do systemu.

Zwracane dane: Sukces: Przekierowanie do strony głównej. Błąd: Powrót do widoku logowania z błędami walidacyjnymi.

Logout (GET)

Parametry: Brak

Opis: Czyści sesję użytkownika i przekierowuje do strony logowania.

Zwracane dane: Przekierowanie do Login.

Profile (GET)

Parametry: Brak

Opis: Wyświetla stronę profilu użytkownika.

Zwracane dane: Widok profilu.

IsLoggedIn (GET)

Parametry: Brak

Opis: Sprawdza, czy użytkownik jest zalogowany.

Zwracane dane: JSON { isLoggedIn: true/false }.

MyEvents (GET)

Parametry: Brak

Opis: Wyświetla listę aktywnych i zarchiwizowanych wydarzeń użytkownika.

Zwracane dane: Widok z listą wydarzeń.

LeaveEvent (POST)

Parametry: int eventId (Identyfikator wydarzenia).

Opis: Umożliwia użytkownikowi opuszczenie wydarzenia przed zakończeniem rejestracji.

Zwracane dane: Przekierowanie do MyEvents.

ResolveTeam (POST)

Parametry: int teamId (Identyfikator drużyny).

Opis: Umożliwia kapitanowi rozwiązanie drużyny przed zakończeniem rejestracji wydarzenia.

Zwracane dane: Przekierowanie do MyEvents.

InvitePlayer (POST)

Parametry: int teamId (Identyfikator drużyny), string email (Adres e-mail zaproszonego użytkownika).

Opis: Wysyła zaproszenie do drużyny do użytkownika.

Zwracane dane: Przekierowanie do MyEvents.

LeaveTeam (POST)

Parametry: int teamId (Identyfikator drużyny).

Opis: Umożliwia użytkownikowi opuszczenie drużyny.

Zwracane dane: Przekierowanie do MyEvents.

RemoveTeamMember (POST)

Parametry: int teamId (Identyfikator drużyny), int memberId (Identyfikator członka drużyny).

Opis: Umożliwia kapitanowi drużyny usunięcie członka zespołu.

Zwracane dane: Przekierowanie do ManageTeam.

GetTeamMembers (POST)

Parametry: int teamId (Identyfikator drużyny).

Opis: Zwraca listę członków drużyny w formacie JSON.

Zwracane dane: Lista członków drużyny.

ManageTeam (GET)

Parametry: int teamId (Identyfikator drużyny).

Opis: Wyświetla widok zarządzania drużyną.

Zwracane dane: Widok ManageTeamViewModel.

2.EventController:

CurrentEvents (GET)

Parametry: Brak

Opis: Pobiera listę aktualnych wydarzeń, które jeszcze się nie odbyły.

Zwracane dane: Widok z listą aktualnych wydarzeń.

ArchivedEvents (GET)

Parametry: Brak

Opis: Pobiera listę archiwalnych wydarzeń, które już się odbyły.

Zwracane dane: Widok z listą archiwalnych wydarzeń.

GetRegistrationStatus

Parametry:

-DateTime? registrationStart: Data rozpoczęcia rejestracji.

-DateTime? registrationEnd: Data zakończenia rejestracji.

Opis: Oblicza status rejestracji dla wydarzenia na podstawie podanych terminów. Sprawdza, czy rejestracja jest otwarta, czy zamknięta i ile czasu pozostało do jej rozpoczęcia lub zakończenia.

Zwracane dane: Status rejestracji jako ciąg znaków.

3.HomeController:

Index (GET)

Parametry: disciplineFilter (string)

Opis: Przekierowuje do widoku wydarzeń z opcjonalnym filtrem dyscyplin.

Zwracane dane: Przekierowanie do akcji CurrentEvents z przekazanym filtrem.

CurrentEvents (GET)

Parametry: disciplineFilter (string)

Opis: Wyświetla listę nadchodzących wydarzeń z możliwością filtrowania według dyscypliny.

Zwracane dane: Widok z listą nadchodzących wydarzeń (EventViewModel).

ArchivedEvents (GET)

Parametry: disciplineFilter (string)

Opis: Wyświetla listę archiwalnych wydarzeń z możliwością filtrowania według dyscypliny.

Zwracane dane: Widok z listą archiwalnych wydarzeń (EventViewModel).

Privacy

Parametry: Brak

Opis: Wyświetla widok polityki prywatności.

Zwracane dane: Widok polityki prywatności.

Error

Parametry: Brak

Opis: Wyświetla widok błędu aplikacji.

Zwracane dane: Widok błędu z modelem ErrorViewModel.

MainPage

Parametry: Brak

Opis: Wyświetla główną stronę aplikacji.

Zwracane dane: Widok strony głównej.

Details (GET)

Parametry: id (int)

Opis: Wyświetla szczegóły indywidualnego wydarzenia.

Zwracane dane: Widok szczegółów wydarzenia (EventDetailsViewModel) lub błąd 404, jeśli wydarzenie nie istnieje.

DetailsTeam (GET)

Parametry: id (int)

Opis: Wyświetla szczegóły drużynowego wydarzenia.

Zwracane dane: Widok szczegółów drużynowego wydarzenia (EventDetailsViewModel) lub błąd 404, jeśli wydarzenie nie istnieje.

RegisterForEvent (POST)

Parametry: eventId (int)

Opis: Rejestruje zalogowanego użytkownika na wydarzenie indywidualne.

Zwracane dane: Przekierowanie do szczegółów wydarzenia z odpowiednim komunikatem (sukces lub błąd).

RedirectToEventDetails

Parametry: eventId (int)

Opis: Przekierowuje użytkownika do odpowiedniego widoku szczegółów (indywidualnego lub drużynowego).

Zwracane dane: Przekierowanie do akcji Details lub DetailsTeam.

RegisterTeamForEvent (POST)

Parametry: eventId (int), teamName (string)

Opis: Rejestruje nową drużynę na wydarzenie drużynowe.

Zwracane dane: Przekierowanie do szczegółów wydarzenia drużynowego z odpowiednim komunikatem (sukces lub błąd).

Participants

Parametry: eventId (int)

Opis: Wyświetla listę uczestników indywidualnych wydarzenia.

Zwracane dane: Widok z listą uczestników wydarzenia.

DetailsRedirect (GET)

Parametry: id (int)

Opis: Przekierowuje do odpowiedniego widoku szczegółów wydarzenia (indywidualnego lub drużynowego).

Zwracane dane: Przekierowanie do akcji Details lub DetailsTeam.

TeamParticipants

Parametry: eventId (int)

Opis: Wyświetla listę drużyn i ich członków biorących udział w wydarzeniu drużynowym.

Zwracane dane: Widok z listą drużyn uczestniczących w wydarzeniu.

AcceptInvitation (POST)

Parametry: id (int)

Opis: Akceptuje zaproszenie użytkownika do drużyny.

Zwracane dane: Przekierowanie do listy zaproszeń z odpowiednim komunikatem (sukces lub błąd).

RejectInvitation (POST)

Parametry: id (int)

Opis: Odrzuca zaproszenie użytkownika do drużyny.

Zwracane dane: Przekierowanie do listy zaproszeń z odpowiednim komunikatem (sukces lub błąd).

TeamInvitations (GET)

Parametry: Brak

Opis: Wyświetla listę zaproszeń do drużyn dla zalogowanego użytkownika.

Zwracane dane: Widok z listą zaproszeń do drużyn.

RespondToInvitation (POST)

Parametry: invitationId (int), accept (bool)

Opis: Obsługuje odpowiedź użytkownika na zaproszenie do drużyny (akceptacja lub odrzucenie).

Zwracane dane: Przekierowanie do listy zaproszeń z odpowiednim komunikatem.

DeleteTeam (POST)

Parametry: teamId (int)

Opis: Usuwa drużynę i jej członków.

Zwracane dane: Przekierowanie do listy wydarzeń użytkownika z odpowiednim komunikatem.

4.HostController:

Index (GET)

Parametry: Brak

Opis: Wyświetla listę aktywnych i nieaktywnych wydarzeń gospodarza.

Zwracane dane: Widok z modelami aktywnych i nieaktywnych wydarzeń.

CreateEvent (GET)

Parametry: Brak

Opis: Wyświetla formularz tworzenia nowego wydarzenia.

Zwracane dane: Widok EventViewModel.

CreateEvent (POST)

Parametry: EventViewModel model

Opis: Tworzy nowe wydarzenie na podstawie przesłanych danych.

Zwracane dane: Sukces: Przekierowanie do Index. Błąd: Widok z błędami walidacji.

DeleteEvent (POST)

Parametry: int id

Opis: Usuwa wydarzenie o podanym ID wraz z powiadomieniami dla uczestników.

Zwracane dane: Przekierowanie do Index.

Details (GET)

Parametry: int id

Opis: Wyświetla szczegóły wydarzenia o podanym ID.

Zwracane dane: Widok szczegółów wydarzenia.

Edit (GET)

Parametry: int id

Opis: Wyświetla formularz edycji dla wydarzenia o podanym ID.

Zwracane dane: Widok EventEditViewModel.

Edit (POST)

Parametry: EventEditViewModel model

Opis: Aktualizuje wydarzenie na podstawie przesłanych danych.

Zwracane dane: Sukces: Przekierowanie do Index. Błąd: Widok z błędami walidacji.

MarkAsInactive (POST)

Parametry: int id

Opis: Oznacza wydarzenie jako nieobstęgiwane.

Zwracane dane: Przekierowanie do Index.

MarkAsUnused (POST)

Parametry: int id

Opis: Oznacza wydarzenie jako nieużywane.

Zwracane dane: Przekierowanie do Index.

MarkAsActive (POST)

Parametry: int id

Opis: Przywraca wydarzenie do aktywnych.

Zwracane dane: Przekierowanie do Index.

5.NotificationsController:

Index (GET)

Parametry: Brak

Opis: Wyświetla listę powiadomień użytkownika zalogowanego na podstawie jego ID z sesji.

Zwracane dane: Widok z listą powiadomień.

MarkAsRead (POST)

Parametry: id (int) - ID powiadomienia

Opis: Oznacza powiadomienie o podanym ID jako przeczytane.

Zwracane dane: Przekierowanie do widoku powiadomień.

GetUnreadCount (GET)

Parametry: Brak

Opis: Zwraca liczbę nieprzeczytanych powiadomień dla aktualnie zalogowanego użytkownika w formacie JSON.

Zwracane dane: JSON { unreadCount: int }.

AddNotification

Parametry: userId (int), message (string)

Opis: Dodaje nowe powiadomienie dla danego użytkownika z określoną treścią.

Zwracane dane: Brak danych zwrotnych.

DeleteNotification (POST)

Parametry: id (int) - ID powiadomienia

Opis: Usuwa powiadomienie o podanym ID, jeśli należy do aktualnie zalogowanego użytkownika.

Zwracane dane: Przekierowanie do widoku powiadomień.

Opis systemu użytkowników:

Użytkownik

Podstawowa rola przypisywana każdemu zarejestrowanemu użytkownikowi. Użytkownicy mogą przeglądać listy nadchodzących i archiwalnych wydarzeń, rejestrować się na zawody indywidualne oraz otrzymywać powiadomienia o istotnych zmianach.

Gospodarz

Rola wybierana podczas rejestracji, przeznaczona dla organizatorów zawodów. Gospodarze mogą tworzyć, edytować i usuwać wydarzenia, zarządzać uczestnikami oraz wysyłać powiadomienia. Mają również możliwość oznaczania wydarzeń jako nieużywane.

Członek drużyny

Rola przypisywana użytkownikowi po dołączeniu do drużyny. Członkowie mogą przeglądać szczegóły swojej drużyny, w tym jej skład i kapitana, oraz otrzymywać powiadomienia dotyczące stanu drużyny.

Kapitan drużyny

Użytkownik tworzący drużynę automatycznie staje się jej kapitanem. Kapitan zarządza drużyną, zaprasza nowych członków, rejestruje drużynę na zawody i ma możliwość przekazania swojej roli lub rozwiązania drużyny.

Charakteryzacja najciekawszych funkcji:

System zaproszeń do drużyn

Użytkownicy mogą zapraszać innych do dołączenia do drużyn. Każde zaproszenie jest rejestrowane w systemie, a zaproszona osoba otrzymuje powiadomienie i możliwość akceptacji lub odrzucenia. Kapitan drużyny zarządza zaproszeniami i decyduje o składzie.

Dynamiczny system powiadomień

Każdy użytkownik otrzymuje powiadomienia o kluczowych wydarzeniach, takich jak zmiany w drużynie, odwołanie wydarzenia czy zbliżający się termin zawodów. Powiadomienia są wyświetlane w dedykowanej sekcji, z możliwością oznaczania ich jako przeczytane lub usuwania.--

Ograniczenia na podstawie stanu rejestracji

System kontroluje, które działania są dostępne w zależności od stanu rejestracji na wydarzenie. Na przykład, po zamknięciu rejestracji uczestnicy nie mogą opuszczać zawodów ani rozwiązywać drużyn.