**aplikacja webowa do przeprowadzania quizów online**

|  |  |  |  |
| --- | --- | --- | --- |
| **Wersja** | **Data utworzenia** | **Data ost. modyfikacji** | **Autor** |
| 2.2 | 11.06.2025 | 20.07.2025 | Kamil Wieczorek  Nr. Albumu: 127 901 |

# Ogólne informacje – Krótki opis działania projektu

QuizApp to aplikacja webowa stworzona w technologii ASP.NET Core MVC, która pozwala użytkownikom na tworzenie, rozwiązywanie i analizowanie quizów online. Użytkownicy po założeniu konta mogą tworzyć własne quizy z pytaniami jednokrotnego wyboru, przeglądać i rozwiązywać quizy stworzone przez innych, a także analizować szczegółowe statystyki dla swoich prac. Aplikacja zawiera również panel administratora z pełnymi uprawnieniami do zarządzania wszystkimi quizami i użytkownikami. Cała aplikacja jest w języku polskim. Aplikacja demonstruje użycie wzorca MVC, systemu uwierzytelniania i autoryzacji, ról użytkowników oraz zaawansowanych operacji na bazie danych.

# Specyfikacja

* Framework: .NET 8
* Platforma: ASP.NET Core MVC
* ORM (Object-Relational Mapping): Entity Framework Core 8
* Baza danych: SQLite
* System użytkowników: ASP.NET Core Identity
* Frontend: HTML, CSS, Bootstrap 5, JavaScript

# Instrukcje pierwszego uruchomienia projektu

1. Jeżeli projekt uruchamiany wykorzystując repozytorium GitHub wtedy:
   1. Sklonuj repozytorium projektu na dysk lokalny.
   2. Otwieramy terminal w miejscu docelowym, do którego chcemy skopiować repozytorium.
   3. Wpisujemy polecenie:

git clone https://github.com/wieczmilkaczorek/quiz-app-kamil-wieczorek-127.git

1. Jeżeli projekt uruchamiany jest przez wersję pliku binarnego – spakowanego folderu zip o nazwie z rozszerzeniem ”QuizApp.zip” wtedy:
   1. Wypakuj folder do pulpitu bądź w katalogu pobrane.
2. Reszta instrukcji przebiega tak samo, dla obydwóch sposobów uruchomienia projektu
3. Otwórz terminal w głównym folderze projektu.
4. Upewnij się, że spełniasz wymogi dotyczące specyfikacji w punkcie 2 w tym celu:
   1. .NET 8
      1. Sprawdź wersję poleceniem: dotnet –version
      2. Jeżeli nie masz zainstalowanego .NET8, to zrób to, przez ten link:  
         <https://dotnet.microsoft.com/download/dotnet/8.0>
5. Przywróć wszystkie pakiety NuGet za pomocą polecenia:
   1. dotnet restore
6. Należy mieć zainstalowane dla EF Core interfejs w wierszu poleceń. Można to zrobić korzystając z polecenia:
   1. Sprawdź czy masz używając komendy: dotnet-ef --version
   2. dotnet tool install –global dotnet-ef
   3. Bądź tylko dla lokalnego projektu używając polecenia poniżej:  
      dotnet tool install dotnet-ef –local
7. Utwórz migrację, która powinna mieć inną nazwę niż InitialCreate, AddCreatedAtFields oraz FactoryInitialCreate. Wykorzystaj do tworzenia migracji poniższe polecenie

dotnet ef migrations add aplikacja

1. Zastosuj wszystkie migracje do bazy danych, aby utworzyć jej schemat:

dotnet ef database update

1. Uruchom aplikację za pomocą polecenia:

dotnet run

1. Otwórz przeglądarkę i wejdź pod adres podany w terminalu (np. `http://localhost:5068`). Jest możliwość skorzystania z skrótu ctrl + lewy przycisk myszy by przejść na adres.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Zawartość wygenerowana przez AI może być niepoprawna.

Przykładowy proces pierwszego uruchomienia znajduje się pod linkiem poniżej tego akapitu oraz na końcu dokumentacji. Link prowadzi do nagrania, w którym projekt jest również pokazywany od strony administratora oraz użytkownika, czyli przedstawia UI.

Link: [Programowanie Zaawansowane](https://1drv.ms/f/c/4dd1859345bb1783/EhrgbWBTgnlOvPS-0lfK__UBPiai8KEELvOVipf-s2qjKA?e=9q0emm)

# Opis struktury projektu

* Aplikacja została zbudowana w oparciu o wzorzec architektoniczny Model-View-Controller (MVC).
* Controllers/: Zawiera klasy kontrolerów, które obsługują żądania przychodzące, przetwarzają dane i decydują, który widok zwrócić.
* Models/: Zawiera klasy C#, które reprezentują strukturę danych w bazie (modele domenowe).
* Views/: Zawiera pliki .cshtml (składnia Razor), które odpowiadają za warstwę prezentacji (interfejs użytkownika).
* ViewModels/: Zawiera klasy modeli widoków, które służą do przekazywania specyficznych, ustrukturyzowanych danych z kontrolerów do widoków.
* Data/: Zawiera klasę ApplicationDbContext odpowiedzialną za konfigurację i komunikację z bazą danych za pomocą Entity Framework Core.
* wwwroot/: Zawiera statyczne pliki, takie jak CSS, JavaScript (quiz-builder.js) i biblioteki klienckie.
* Areas/Identity/: Zawiera wygenerowane pliki dla systemu logowania i rejestracji.

# Modele

* Quiz: Główny model reprezentujący pojedynczy quiz.
  + Id (int): Klucz główny.
  + Title (string): Tytuł quizu. Wymagany, max 100 znaków.
  + Description (string): Opis quizu. Max 500 znaków.
  + AuthorId (string): Klucz obcy do autora (ApplicationUser).
  + CreatedAt (DateTime): Data i czas utworzenia quizu.
* Question: Reprezentuje pojedyncze pytanie w quizie.
  + Id (int): Klucz główny.
  + Text (string): Treść pytania. Wymagane.
  + Points (int): Liczba punktów za poprawną odpowiedź. Wymagane, zakres 1-100.
  + QuizId (int): Klucz obcy do quizu.
* Answer: Reprezentuje pojedynczą odpowiedź na pytanie.
  + Id (int): Klucz główny.
  + Text (string): Treść odpowiedzi. Wymagane.
  + IsCorrect (bool): Flaga wskazująca, czy odpowiedź jest poprawna.
  + QuestionId (int): Klucz obcy do pytania.
* ApplicationUser: Rozszerzenie standardowego użytkownika Identity. Przechowuje powiązanie z quizami stworzonymi przez użytkownika.
  + CreatedAt (DateTime): Data i czas rejestracji użytkownika.
* QuizAttempt: Zapisuje informację o podejściu użytkownika do quizu.
  + Id (int): Klucz główny.
  + QuizId (int): Klucz obcy do rozwiązywanego quizu.
  + UserId (string): Klucz obcy do użytkownika, który rozwiązał quiz.
  + Score (int): Zdobyta liczba punktów.
  + AttemptedOn (DateTime): Data i czas podejścia.
* UserAnswer: Zapisuje konkretną odpowiedź wybraną przez użytkownika w danym podejściu.
  + Id (int): Klucz główny.
  + QuizAttemptId (int): Klucz obcy do próby rozwiązania quizu.
  + QuestionId (int): Klucz obcy do pytania.
  + SelectedAnswerId (int): Klucz obcy do wybranej odpowiedzi.

# Wylistowane kontrolery z metodami

* QuizController: Zarządza tworzeniem, wyświetlaniem i rozwiązywaniem quizów.
  + Create [GET]: Wyświetla dynamiczny formularz do tworzenia nowego quizu.
  + Create [POST]: Przetwarza dane z formularza i zapisuje nowy quiz w bazie danych.
  + Index [GET]: Wyświetla listę wszystkich dostępnych quizów (dla zalogowanych) lub komunikat o konieczności logowania (dla niezalogowanych).
  + MyQuizzes [GET]: Wyświetla listę quizów stworzonych przez zalogowanego użytkownika.
  + Take [GET]: Wyświetla stronę z pytaniami quizu do rozwiązania.
  + Submit [POST]: Odbiera odpowiedzi użytkownika, oblicza wynik i zapisuje próbę w bazie.
  + Results [GET]: Wyświetla stronę z wynikiem po rozwiązaniu quizu.
  + Delete [GET]: Wyświetla stronę z potwierdzeniem usunięcia quizu.
  + DeleteConfirmed [POST]: Usuwa quiz z bazy danych.
* AdminController: Zarządza panelem administratora (tylko dla roli Admin).
  + Index [GET]: Wyświetla listę wszystkich quizów w systemie z opcjami zarządzania.
  + EditQuiz [GET]: Wyświetla formularz edycji dowolnego quizu.
  + EditQuiz [POST]: Zapisuje zmiany w edytowanym quizie.
  + DeleteQuiz [GET]: Wyświetla stronę potwierdzenia usunięcia quizu.
  + DeleteQuizConfirmed [POST]: Usuwa wybrany quiz z bazy danych.
  + Users [GET]: Wyświetla listę wszystkich użytkowników z informacjami o rolach i statystykach.
* StatisticsController: Odpowiada za wyświetlanie statystyk.
  + Details [GET]: Oblicza i wyświetla szczegółowe statystyki dla danego quizu (liczba podejść, średni wynik, rozkład odpowiedzi). Dostępna tylko dla autora quizu.
* Inne: HomeController (strona główna), kontrolery z Identity (obsługa kont).

# Opis systemu użytkowników ze wszystkimi jego komponentami

* Aplikacja wykorzystuje ASP.NET Core Identity do zarządzania użytkownikami z systemem ról.
* Role w systemie:
  + Zwykły użytkownik: Domyślna rola dla wszystkich zarejestrowanych użytkowników.
  + Administrator: Rola z pełnymi uprawnieniami do zarządzania wszystkimi quizami i użytkownikami.
* Domyślne konto administratora: „admin@quizapp.com” / „Admin123!” (tworzone automatycznie przy pierwszym uruchomieniu).
* Rejestracja jest otwarta dla każdego. Po zalogowaniu użytkownik może tworzyć i rozwiązywać quizy.
* Zastosowano **personalizację danych**:
  + Użytkownik na stronie "Moje Quizy" widzi tylko te quizy, których jest autorem.
  + Statystyki dla danego quizu może przeglądać wyłącznie jego autor.
  + Administrator ma dostęp do panelu zarządzania wszystkimi quizami i użytkownikami.
* Dostęp do kluczowych funkcjonalności (tworzenie, rozwiązywanie, statystyki) jest chroniony atrybutem `[Authorize]`.
* Panel administratora jest dostępny tylko dla użytkowników z rolą "Admin" (`[Authorize(Roles = "Admin")]`).

# Krótka charakterystyka najciekawszych funkcjonalności

* Dynamiczne tworzenie quizów: Formularz do tworzenia quizu został zbudowany z użyciem JavaScript, co pozwala użytkownikowi na dynamiczne dodawanie i usuwanie dowolnej liczby pytań i odpowiedzi bez przeładowywania strony. Dane z tak złożonego formularza są poprawnie mapowane na `ViewModel` i przetwarzane przez kontroler.
* Zaawansowane statystyki: Aplikacja nie tylko zapisuje wyniki, ale także wykonuje zaawansowane operacje agregujące bezpośrednio w bazie danych za pomocą LINQ (`GroupBy`, `AverageAsync`, `CountAsync`). Dzięki temu autor quizu otrzymuje wartościowe informacje zwrotne, np. procentowy rozkład popularności każdej z odpowiedzi, co pozwala ocenić jakość i trudność pytań.
* Panel administratora: Kompleksowy system zarządzania z możliwością edycji i usuwania wszystkich quizów, niezależnie od autora. Administrator może także przeglądać listę wszystkich użytkowników z ich rolami i statystykami.
* Kontrola dostępu: Niezalogowani użytkownicy widzą informację o konieczności logowania zamiast listy quizów, co zapewnia bezpieczeństwo danych.
* Pełna lokalizacja: Wszystkie elementy interfejsu, komunikaty błędów i strony Identity są przetłumaczone na język polski.
* Automatyczna inicjalizacja: System automatycznie tworzy rolę administratora i domyślne konto przy pierwszym uruchomieniu.

# Materiały dodatkowe:

* Link do nagrania pierwsze uruchomienie projektu:  
  <https://1drv.ms/f/c/4dd1859345bb1783/EhrgbWBTgnlOvPS-0lfK__UBPiai8KEELvOVipf-s2qjKA?e=9q0emm>