Aplikacja do zarządzania ocenami

# Opis aplikacji

Nasza aplikacja umożliwia efektywne zarządzanie danymi edukacyjnymi, zapewniając zabezpieczony dostęp do funkcji w zależności od roli użytkownika. Umożliwia rejestrację, logowanie i zarządzanie studentami, ocenami oraz przedmiotami, ułatwiając tym samym codzienną administrację edukacyjną. Dzięki klarownemu interfejsowi i zaawansowanej logice biznesowej, nasze rozwiązanie wspiera efektywność zarządzania edukacją na każdym poziomie.

# Wykorzystane technologie

Aplikacja korzysta z .NET w wersji 8.0.1. Dane są przechowywane w wbudowanej do aplikacji Visual Studio bazie danych

Projekt korzysta z poniższych bibliotek:

* Microsoft.AspNetCore.Components.QuickGrid.EntityFrameworkAdapter (8.0.4)
* Microsoft.AspNetCore.Diagnostics.Entity FrameworkCore (8.0.1)
* Microsoft.AspNetCore.Identity.Entity FrameworkCore (8.0.4)
* Microsoft.AspNetCore.Identity.Ul (8.0.4)
* Microsoft.EntityFrameworkCore (8.0.6)
* Microsoft.Entity FrameworkCore.Sqlite (8.0.4)
* Microsoft.EntityFrameworkCore.SqlServer (8.0.4)
* Microsoft.EntityFrameworkCore.Tools (8.0.4)
* Microsoft.VisualStudio.Web.CodeGeneration.Design (8.0.2)

# Instrukcja pierwszego uruchomienia

Przed uruchomieniem aplikacji należy upewnić się, że masz zainstalowane najnowsze narzędzia deweloperskie .NET 8.0 i Visual Studio. Następnie należy wejść w Narzędzia -> Menadźer Pakietów NuGet - > Zarządzaj pakietami NuGet rozwiązania i sprawdzić czy posiadamy zainstalowane wszystkie biblioteki wypisane w punkcie Wykorzystane technologie. Teraz w Konsoli menadżera pakietów należy wpisać komendę update-database oraz uruchomić aplikację.

Z powodu błędu, który nie pozwala na zalogowanie na zseedowane konta admin o loginie [superadmin@gmail.com](mailto:superadmin@gmail.com) i haśle 123Pa$$word, należy zarejestrować konto użytkownika, zalogować się na nie oraz zmienić jego uprawnienia w aplikacji w widoku userroles, użyć przycisku manage oraz dodać rolę moderator. Następnie trzeba się wylogować oraz zalogować ponownie. Na czas przed zmianą uprawnień kod odpowiadający za blokowanie użytkowników bez odpowiedniej roli został wpisany w komentarz w pliku UserRolesController.cs na linii 20 oraz 70.

# Opis struktury projektu

# Modele

1. Course - Model Course reprezentuje przedmiot nauczania w systemie edukacyjnym, zawierający informacje takie jak nazwe, ID, punktację oraz kolekcję Enrollments
2. Enrollment - Model Enrollment reprezentuje zapis studenta na konkretne zajęcia lub kursy. Zawiera informacje takie jak ID, CourseID, StudentID, ocenę.
3. Manage - Model Manage może odnosić się do różnych zadań zarządzania, ale najczęściej jest używany w kontekście zarządzania użytkownikami, rolami i uprawnieniami w systemie
4. Student - Model Student reprezentuje informacje o studencie, takie jak nazwisko, imię, full name czyli zmienną łącząca nazwisko i imię, kolekcję enrollments oraz datę zapisu
5. UserRolesViewModel - UserRolesViewModel jest modelem używanym do reprezentacji danych potrzebnych do zarządzania rolami użytkowników. Zawiera ID, username oraz email oraz kolekcję Roles, która zawiera przypisane role

# Kontrolery

Kontroler CoursesController zarządza operacjami CRUD dla kursów. Metoda Index wyświetla listę wszystkich kursów. Details pokazuje szczegóły wybranego kursu. Create zezwala na dodanie nowego kursu, Edit na edycję, a Delete na usunięcie. Metody POST zabezpieczone są przed atakami typu "overposting". Wszystkie akcje zwracają widoki lub przekierowania, a CourseExists sprawdza istnienie kursu w bazie danych.

Kontroler EnrollmentsController zarządza operacjami CRUD dla zapisów (enrollmentów). Index wyświetla listę wszystkich zapisów. Details pokazuje szczegóły wybranego zapisu. Create umożliwia dodanie nowego zapisu, Edit edytowanie, a Delete usuwanie. Metody POST są zabezpieczone przed atakami typu "overposting". Wszystkie akcje zwracają widoki lub przekierowania, a EnrollmentExists sprawdza istnienie zapisu w bazie danych. Akcje są dostępne tylko dla ról administratora, superadministratora i moderatora.

Kontroler HomeController obsługuje podstawowe akcje strony głównej aplikacji. Index zwraca widok strony głównej. Privacy zwraca widok strony prywatności. Error obsługuje błędy i zwraca widok błędu wraz z odpowiednim identyfikatorem żądania.

Kontroler RoleManagerController zarządza rolami użytkowników. Jest zabezpieczony atrybutem Authorize, co oznacza, że wymaga uwierzytelnienia użytkownika. Index (GET) zwraca widok listy wszystkich ról dla użytkownika z uprawnieniami "Admin", "SuperAdmin" lub "Moderator". AddRole (POST) pozwala na dodanie nowej roli, ale dostępna jest tylko dla użytkowników z uprawnieniami "SuperAdmin" lub "Moderator". Dodawanie roli odbywa się poprzez metodę CreateAsync klasy RoleManager<IdentityRole>. Kontroler obsługuje operacje związane z zarządzaniem rolami użytkowników, zapewniając odpowiednie zabezpieczenia dostępu do poszczególnych funkcji w zależności od roli użytkownika.

Kontroler StudentsController zarządza operacjami CRUD dla studentów. Index (GET) wyświetla listę studentów, umożliwiając sortowanie i wyszukiwanie. Details (GET) pokazuje szczegóły wybranego studenta. Create (GET) i Create (POST) umożliwiają dodanie nowego studenta. Edit (GET) i Edit (POST) pozwalają edytować istniejące dane studenta. Delete (GET) i DeleteConfirmed (POST) usuwają studenta.

Kontroler jest zabezpieczony atrybutem Authorize, co oznacza, że dostęp do wszystkich akcji mają tylko użytkownicy z rolami "Admin", "SuperAdmin" lub "Moderator". Akcje POST są zabezpieczone przed atakami typu "overposting" poprzez użycie atrybutu ValidateAntiForgeryToken.

Każda akcja zwraca odpowiedni widok lub wykonuje przekierowanie do akcji Index w przypadku sukcesu. Metoda StudentExists sprawdza istnienie studenta w bazie danych przed próbą edycji lub usunięcia.

Kontroler UserRolesController zarządza rolami użytkowników. Jest zabezpieczony atrybutem Authorize, co oznacza, że wymaga uwierzytelnienia użytkownika. Index (GET) wyświetla listę użytkowników oraz ich role dla administratorów, superadministratorów lub moderatorów. Manage (GET) pozwala na zarządzanie rolami wybranego użytkownika, pobierając dostępne role z RoleManager<IdentityRole> i sprawdzając przynależność użytkownika do poszczególnych ról. Manage (POST) aktualizuje role użytkownika na podstawie wybranych opcji, dostępny tylko dla superadministratorów i moderatorów.

Kontroler obsługuje operacje związane z zarządzaniem rolami użytkowników, zapewniając odpowiednie zabezpieczenia dostępu do poszczególnych funkcji w zależności od roli użytkownika. Metody POST są zabezpieczone przed atakami typu "overposting" poprzez użycie atrybutu ValidateAntiForgeryToken.

Akcje kontrolera obejmują zarządzanie rolami użytkowników, usuwanie i dodawanie ról, oraz są dostępne tylko dla uprawnionych użytkowników.

# Opis systemu

Obecnie w systemie znajduje się podział na 4 role, basic, admin, superadmin oraz moderator, wszystkie role znajdują się w podstronie RoleManager. Wszyscy użytkownicy są wypisani w podstronie User Roles wraz z ich rolami. W celu zmiany roli dla wybranego użytkownika, użytkownik musi zalogować się na konto moderatora lub superadmina, wejść w podstronę User Role, wybrać, która rola ma zostać nadana lub odebrana użytkownikowi.

Niezalogowany użytkownik ma dostęp jedynie do podstron Students, Courses oraz Enrollments. Zalogowany użytkownik z rolą basic może dodatkowo wejść w szczegóły poszczególnych rekordów na każdej z wypisanych wcześniej podstron. Rola admin pozwala na dodawania, zmianę oraz usuwanie rekordów w powyższych podstronach oraz pozwala na wejście w podstrony Role Manager oraz User Roles. Rola Moderator i Super Admin mają pełny zakres uprawnień.