

BookManagement

Autor Projektu:

Dawid Czepułkowski

Opis Działania Projektu:

BookManagement to aplikacja webowa umożliwiająca użytkownikom zarządzanie książkami. Użytkownicy mogą rejestrować konta, logować się, dodawać książki do swojej kolekcji, oceniać książki, przeszukiwać dostępne książki oraz usuwać książki z ich kolekcji tym samym udostępniając je z powrotem innym użytkownikom.

Specyfikacja Wykorzystanych Technologii:

- .NET 6
- ASP.NET Core MVC
- Entity Framework Core
- SQLite (baza danych)
- HTML, CSS (Bootstrap)
- JavaScript

Instrukcje Pierwszego Uruchomienia Projektu:

1. Sklonuj repozytorium projektu:

```
git clone https://github.com/DavvidC/BookManagement.git
cd BookManagement
```

2. Otwórz projekt w Visual Studio:

- Uruchom Visual Studio.
- Wybierz File -> Open -> Project/Solution.
- Wskaż na plik .sln projektu w folderze BookManagement.

3. Przywróć pakiety NuGet:

- W Visual Studio przejdź do Tools -> NuGet Package Manager -> Package Manager Console.
- Wykonaj polecenie:

```
Update-Package
```

4. Wykonaj migracje bazy danych:

- W Package Manager Console wykonaj polecenie:

```
Update-Database
```

5. Uruchom aplikację:

- Naciśnij F5 lub kliknij Start w Visual Studio.

Opis Struktury Projektu:

Projekt wykorzystuje wzorzec architektoniczny MVC (Model-View-Controller).

- **Model:** Reprezentuje dane aplikacji i logikę biznesową.
- **View:** Odpowiada za wyświetlanie danych użytkownikowi.
- **Controller:** Przechwytuje żądania użytkownika, przetwarza je i zwraca odpowiednie widoki.

Modele:

1. Book

- Opis: Reprezentuje książkę.
- Pola:
 - `Id (int)`: Unikalny identyfikator książki.
 - `Title (string)`: Tytuł książki (wymagane).
 - `Author (string)`: Autor książki (wymagane).
 - `Description (string)`: Opis książki.
 - `IsPublic (bool)`: Określa, czy książka jest publiczna.
 - `UserId (int?)`: Id użytkownika przypisanego do książki (może być null).
 - `User (User)`: Użytkownik przypisany do książki.
 - `AverageRating (double)`: Średnia ocena książki.

2. User

- Opis: Reprezentuje użytkownika.
- Pola:
 - `Id (int)`: Unikalny identyfikator użytkownika.
 - `FirstName (string)`: Imię użytkownika (wymagane).
 - `LastName (string)`: Nazwisko użytkownika (wymagane).
 - `Username (string)`: Nazwa użytkownika (wymagane).
 - `Password (string)`: Hasło użytkownika (wymagane).
 - `Books (ICollection<Book>)`: Kolekcja książek przypisanych do użytkownika.

3. Rating

- Opis: Reprezentuje ocenę książki.
- Pola:
 - `Id (int)`: Unikalny identyfikator oceny.
 - `Value (int)`: Wartość oceny (1-5).
 - `BookId (int)`: Id książki, której ocena dotyczy.
 - `Book (Book)`: Książka przypisana do oceny.

Kontrolery i Metody:

1. HomeController

- **Index**
 - HTTP GET
 - Parametry: brak
 - Opis: Wyświetla stronę główną z losowymi książkami.
 - Zwracane dane: Widok `Index`
- **Register**
 - HTTP GET
 - Parametry: brak
 - Opis: Wyświetla formularz rejestracji użytkownika.
 - Zwracane dane: Widok `Register`
- **Register**
 - HTTP POST
 - Parametry: `User user`
 - Opis: Przetwarza dane rejestracyjne użytkownika.
 - Zwracane dane: Przekierowanie do `Login` lub widok `Register` z błędami.
- **Login**
 - HTTP GET
 - Parametry: brak
 - Opis: Wyświetla formularz logowania.
 - Zwracane dane: Widok `Login`
- **Login**
 - HTTP POST
 - Parametry: `string username, string password`
 - Opis: Przetwarza dane logowania użytkownika.
 - Zwracane dane: Przekierowanie do `Index` lub widok `Login` z błędami.
- **Logout**
 - HTTP GET
 - Parametry: brak
 - Opis: Wylogowuje użytkownika.
 - Zwracane dane: Przekierowanie do `Index`
- **AddToMyBooks**
 - HTTP POST
 - Parametry: `int id`
 - Opis: Dodaje książkę do kolekcji użytkownika.
 - Zwracane dane: Przekierowanie do `Index`

2. BookController

- **Index**
 - HTTP GET
 - Parametry: `string searchString`
 - Opis: Wyświetla listę wszystkich książek (z opcją szukania).
 - Zwracane dane: Widok `Index`
- **MyBooks**
 - HTTP GET
 - Parametry: `string searchString`
 - Opis: Wyświetla listę książek użytkownika (z opcją szukania).
 - Zwracane dane: Widok `MyBooks`

- **Create**
 - HTTP GET
 - Parametry: brak
 - Opis: Wyświetla formularz dodawania książki.
 - Zwracane dane: Widok `Create`
- **Create**
 - HTTP POST
 - Parametry: `Book book`
 - Opis: Przetwarza dane nowej książki.
 - Zwracane dane: Przekierowanie do `Index` lub widok `Create` z błędami.
- **Details**
 - HTTP GET
 - Parametry: `int id`
 - Opis: Wyświetla szczegóły książki.
 - Zwracane dane: Widok `Details`
- **AddToMyBooks**
 - HTTP POST
 - Parametry: `int id`
 - Opis: Dodaje książkę do kolekcji użytkownika.
 - Zwracane dane: Przekierowanie do `Index`
- **RemoveFromMyBooks**
 - HTTP POST
 - Parametry: `int id`
 - Opis: Usuwa książkę z kolekcji użytkownika.
 - Zwracane dane: Przekierowanie do `MyBooks`
- **ReleaseBook**
 - HTTP POST
 - Parametry: `int id`
 - Opis: Zwalnia książkę, aby była dostępna dla innych użytkowników.
 - Zwracane dane: Przekierowanie do `MyBooks`
- **AddRating**
 - HTTP POST
 - Parametry: `int bookId, int rating`
 - Opis: Dodaje ocenę do książki.
 - Zwracane dane: Przekierowanie do `Index`

System Użytkowników:

- Role: Brak zaawansowanego systemu ról. Użytkownicy są jedynie podzieleni na zalogowanych i niezalogowanych.
- Zalogowani użytkownicy mogą:
 - Dodawać książki do swojej kolekcji.
 - Usuwać książki z swojej kolekcji.
 - Oceniać książki.
 - Przeglądać swoje książki.
- Niezalogowani użytkownicy mogą:
 - Przeglądać dostępne książki.
 - Rejestrować się i logować.

Krótką Charakterystyka Najciekawszych Funkcjonalności:

- **System Oceny Książek:** Użytkownicy mogą oceniać książki, a średnia ocena jest wyświetlana na liście książek.
- **Szukaj Książek:** Możliwość przeszukiwania książek po tytule lub autorze.
- **Indywidualna Kolekcja Książek:** Zalogowani użytkownicy mogą dodawać książki do swojej prywatnej kolekcji i nimi zarządzać.

Uwagi Końcowe:

Aplikacja BookManagement spełnia wymagania projektu, oferując funkcjonalności CRUD dla książek, system użytkowników, możliwość oceniania książek oraz wyszukiwania ich po tytule i autorze.