Dokumentacja techniczna projektu

Tytuł projektu: Serwis Książkowy

Krótki opis działania projektu

Serwis książkowy to platforma społecznościowa umożliwiająca użytkownikom zarządzanie swoją wirtualną biblioteką oraz odkrywanie nowych książek i autorów. Serwis posiada system rejestracji i logowania, dzięki któremu każdy użytkownik ma dostęp do swojego konta, gdzie może przeglądać dostępne książki, oceniać je, pisać recenzje oraz ustawiać ich status jako „Przeczytane” lub „Do przeczytania”. Oferta książek może być filtrowana według kryteriów takich jak gatunek, ocena, autor czy data wydania. Platforma umożliwia również śledzenie ulubionych autorów.

Autor projektu

Albert Kaźmierczak

Specyfikacja wykorzystanych technologii

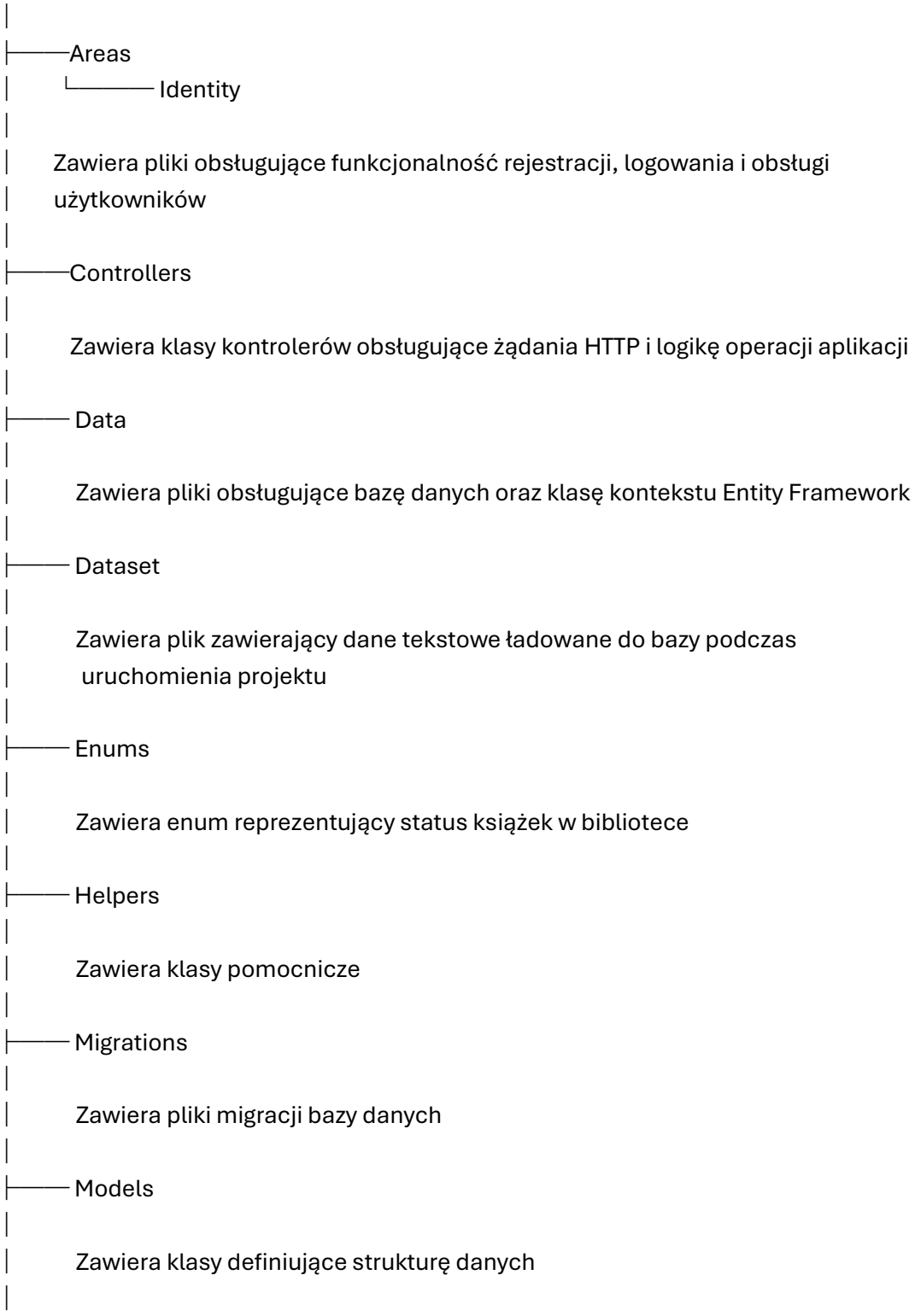
- .NET 8.0
- Język C#
- Microsoft SQL Server LocalDB
- AspNetCore 8.0.11
- Entity Framework 8.0.11
- AspNet Identity 8.0.11
- Bootstrap 5.1
- CsvHelper 33.0.1

Instrukcje pierwszego uruchomienia projektu

- Otwórz projekt w Visual Studio 2022
- Upewnij się że SQL Server LocalDB jest zainstalowane (powinno być zainstalowane razem z VS2022)

- dotnet ef database update w terminalu lub Update-Database w konsoli Menedżera Pakietów
- Uruchom projekt
- Poczekaj aż program zasili bazę danych

Opis struktury projektu



├─── ViewModels
│
│ Zawiera klasy przeznaczone do przekazywania danych między kontrolerami a widokami
│
├─── Views
│
│ Zawiera widoki aplikacji
│
└─── wwwroot

Zawiera pliki statyczne aplikacji

Wszystkie modele

- AppUser

  Model reprezentujący użytkownika aplikacji. Dziedziczy po IdentityUser z ASP.NET Identity, co umożliwia zarządzanie użytkownikami, rolami i uwierzytelnianiem.

  public ICollection<UserLibrary> UserLibraries

  Kolekcja książek powiązanych z użytkownikami

  public ICollection<Review> Reviews

  Kolekcja recenzji powiązanych z użytkownikami

  public ICollection<FavouriteAuthor> FavouriteAuthors

  Kolekcja ulubionych autorów powiązanych z użytkownikami

- Author

  Model reprezentujący autora książek

  public int AuthorId

  Klucz główny.

  [StringLength(150, ErrorMessage = "Author name cannot be longer than 150 characters")]

  public string Name

  Imię i nazwisko autora. Ograniczenia: maksymalna długość 150 znaków.

  public ICollection<FavouriteAuthor> FavouriteAuthors

  Kolekcja ulubionych autorów powiązanych z użytkownikami.

- Book

  Model reprezentujący książkę.

  public int BookId

  Klucz główny książki.

  [Required(ErrorMessage = "ISBN is required")]

  [StringLength(13, MinimumLength = 13, ErrorMessage = "ISBN has to be 13 characters long")]

  [Display(Name = "ISBN")]

  public string Isbn

  Numer ISBN książki. Ograniczenia: dokładnie 13 znaków. Okładka książki jest pobierana po numerze ISBN.

  [Required(ErrorMessage = "Title is required")]

  [StringLength(200, ErrorMessage = "Title cannot be longer than 200 characters")]

  [Display(Name = "Title")]

  public string Title

  Tytuł książki. Ograniczenia: maksymalna długość 200 znaków.

  [Required(ErrorMessage = "Author is required")]

  [Display(Name = "Author")]

  public int AuthorId

  Identyfikator autora książki.

  public Author? Author

  Obiekt autora książki.

  [Required(ErrorMessage = "Genre is required")]

  [Display(Name = "Genre")]

  public int GenreId

  Identyfikator gatunku książki.

public Genre? Genre
Obiekt gatunku książki.
[DisplayFormat(DataFormatString = "{0:dd MMMM, yyyy}")]
[Display(Name = "Publication Date")]
public DateTime PublicationDate
Data publikacji książki.
[Range(0, 5, ErrorMessage = "Rating must be between 0 and 5")]
[Display(Name = "Rating")]
public float? Rating
Ocena książki. Ograniczenia: wartość między 0 a 5.
public ICollection<Review> Reviews
Kolekcja recenzji książki.
public ICollection<UserLibrary> UserLibraries
Kolekcja powiązań książki z bibliotekami użytkowników.

- FavouriteAuthor

Model reprezentujący powiązanie między użytkownikiem a ulubionym autorem.
public string UserId
Identyfikator użytkownika.
public AppUser User
Obiekt użytkownika.
public int AuthorId
Identyfikator autora.
public Author Author
Obiekt autora.

- Genre

Model reprezentujący gatunek literacki.
public int GenreId
Klucz główny.
[Required(ErrorMessage = "Genre name is required")]
[StringLength(50, ErrorMessage = "Genre name cannot be longer than 50 characters")]
[Display(Name = "Genre Name")]
public string Name
Nazwa gatunku. Ograniczenia: maksymalna długość 50 znaków.
public ICollection<Book> Books
Kolekcja książek przypisanych do gatunku.

- Review

Model reprezentujący recenzję książki.
public int ReviewId
Klucz główny.
public string? UserId
Identyfikator użytkownika, który napisał recenzję.
public AppUser? User
Obiekt użytkownika.
[Required(ErrorMessage = "Book ID is required")]
[Display(Name = "Book ID")]
public int BookId
Identyfikator książki, której dotyczy recenzja.
public Book Book
Obiekt książki.
[Required(ErrorMessage = "Rating is required")]
[Range(0, 5, ErrorMessage = "Rating must be between 0 and 5")]
[Display(Name = "Rating")]
public float Rating
Ocena książki. Ograniczenia: wartość między 0 a 5.
[Required(ErrorMessage = "Review text is required")]
[StringLength(1000, ErrorMessage = "Review text cannot be longer than 1000 characters")]
[Display(Name = "Review Text")]
public string ReviewText
Tekst recenzji. Ograniczenia: maksymalna długość 1000 znaków.
[Required(ErrorMessage = "Creation date is required")]
[Display(Name = "Created At")]
[DisplayFormat(DataFormatString = "{0:dd MMMM, yyyy}")]
public DateTime CreatedAt
Data utworzenia recenzji.

- UserLibrary

  Model reprezentujący powiązanie między użytkownikiem a książką w jego bibliotece.

  ```
  [Required(ErrorMessage = "User ID is required")]
  [Display(Name = "User ID")]
  public string UserId
  ```
  Identyfikator użytkownika.
  ```
  public AppUser User
  ```
  Obiekt użytkownika.
  ```
  [Required(ErrorMessage = "Book ID is required")]
  [Display(Name = "Book ID")]
  public int BookId
  ```
  Identyfikator książki.
  ```
  public Book Book
  ```
  Obiekt książki.
  ```
  [Required(ErrorMessage = "Status is required")]
  [Display(Name = "Status")]
  public Status Status
  ```
  Status książki w bibliotece użytkownika.
  ```
  [Required(ErrorMessage = "Added date is required")]
  [Display(Name = "Added At")]
  [DisplayFormat(DataFormatString = "{0:dd MMMM, yyyy}")]
  public DateTime AddedAt
  ```
  Data dodania książki do biblioteki użytkownika.

Kontrolery wraz z metodami

1. AuthorsController

Kontroler zarządzający autorami.

- Index (GET):
  - Wyświetla listę książek autora. Tworzy obiekt AuthorBookViewModel składający się z obiektu AuthorViewModel oraz listy obiektów BookViewModel. AuthorViewModel zawiera autora oraz bool IsFollowed zawierającą informację, czy dany autor jest śledzony przez użytkownika. BookViewModel zawiera książkę Book oraz bool IsInLibrary i Status, zawierające informacje o relacji książki z biblioteką użytkownika. Zawiera paginację.
  - Parametry: int authorId, int page = 1, int pageSize = 10
  - Zwraca: Przekierowanie do strony Authors/Index.cshtml wraz z obiektem AuthorBookViewModel.
- Follow (POST):
  - Dodaje autora do ulubionych użytkownika. Dostępne dla zalogowanych użytkowników.
  - Parametry: int authorId
  - Zwraca: Przekierowanie do poprzedniej strony.
- Unfollow (POST):
  - Usuwa autora z ulubionych użytkownika. Dostępne dla zalogowanych użytkowników.

- Parametry: int authorId
- Zwraca: Przekierowanie do poprzedniej strony.

- FollowedAuthors (GET):
  - Wyświetla listę ulubionych autorów użytkownika wraz z ich najnowszą książką. Dostępne dla zalogowanych użytkowników. Zawiera paginację.
  - Parametry: int page = 1, int pageSize = 10
  - Zwraca: Widok z listą ulubionych autorów.
- Create (GET):
  - Wyświetla formularz tworzenia nowego autora. Dostępne dla administratora.
  - Zwraca: Widok formularza tworzenia.
- Create (POST):
  - Tworzy nowego autora. Dostępne dla administratora.
  - Parametry: obiekt Author
  - Zwraca: Przekierowanie do strony głównej.
- Edit (GET):
  - Wyświetla formularz edycji autora. Dostępne dla administratora.
  - Parametry: int id
  - Zwraca: Widok formularza edycji.
- Edit (POST):
  - Aktualizuje dane autora. Dostępne dla administratora.
  - Parametry: int id, Author author
  - Zwraca: Przekierowanie widoku autora.
- Delete (GET):
  - Wyświetla formularz usunięcia autora. Dostępne dla administratora.
  - Parametry: int id
  - Zwraca: Widok formularza.
- DeleteConfirmed (POST):
  - Usuwa autora. Dostępne dla administratora.
  - Parametry: int id
  - Zwraca: Przekierowanie do listy książek.

2. BooksController

Kontroler zarządzający książkami.

- Index (GET):
  - Wyświetla listę 10 książek posortowanych według ocen malejąco. Zawiera paginację.
  - Parametry: int page = 1, int pageSize = 10
  - Zwraca: Widok z listą książek.
- Details (GET):
  - Wyświetla szczegóły książki.
  - Parametry: int id
  - Zwraca: Widok z szczegółami książki.
- Create (GET):
  - Wyświetla formularz tworzenia nowej książki. Dostępne dla administratora.
  - Zwraca: Widok formularza.
- Create (POST):
  - Tworzy nową książkę. Dostępne dla administratora.
  - Parametry: Book book
  - Zwraca: Przekierowanie do listy książek.
- Edit (GET):
  - Wyświetla formularz edycji książki. Dostępne dla administratora.
  - Parametry: int id
  - Zwraca: Widok edycji książki.
- Edit (POST):
  - Aktualizuje dane książki. Dostępne dla administratora.
  - Parametry: int id, Book book
  - Zwraca: Przekierowanie do szczegółów książki.
- Delete (GET):
  - Wyświetla formularz usunięcia książki. Dostępne dla administratora.
  - Parametry: int id
  - Zwraca: Widok formularza usunięcia książki.

- DeleteConfirmed (POST):
    - Usuwa książkę. Dostępne dla administratora.
    - Parametry: int id
    - Zwraca: Przekierowanie do listy książek.
- AddReview (POST):
    - Dodaje recenzję wraz z oceną do książki. Ogólna ocena książki zostaje przeliczona po dodaniu recenzji. Dostępne dla zalogowanego użytkownika.
    - Parametry: int bookId, string reviewText, float rating
    - Zwraca: Przekierowanie do szczegółów książki.
- ListBooks (GET):
    - Wyświetla kompaktową listę książek z odnośnikami do edycji, detali oraz usunięcia. Zawiera paginację. Dostępne dla administratora.
    - Parametry: int page = 1, int pageSize = 30
    - Zwraca: Widok z listą książek.

3. GenresController

Kontroler zarządzający gatunkami książek.

- Index (GET):
    - Wyświetla listę gatunków. Dostępne dla administratora.
    - Zwraca: Widok z listą gatunków.
- Details (GET):
    - Wyświetla szczegóły gatunku. Dostępne dla administratora.
    - Parametry: int id
    - Zwraca: Widok z szczegółami gatunku.
- Create (GET):
    - Wyświetla formularz tworzenia nowego gatunku. Dostępne dla administratora.
    - Zwraca: Widok formularza.
- Create (POST):
    - Tworzy nowy gatunek. Dostępne dla administratora.
    - Parametry: Genre genre
    - Zwraca: Przekierowanie do listy gatunków.

- Edit (GET):
    - Wyświetla formularz edycji gatunku. Dostępne dla administratora.
    - Parametry: int id
    - Zwraca: Widok formularza edycji.
- Edit (POST):
    - Aktualizuje dane gatunku. Dostępne dla administratora.
    - Parametry: int id, Genre genre
    - Zwraca: Przekierowanie do listy gatunków.
- Delete (GET):
    - Wyświetla formularz usunięcia gatunku. Dostępne dla administratora.
    - Parametry: int id
    - Zwraca: Widok formularza.
- DeleteConfirmed (POST):
    - Usuwa gatunek. Dostępne dla administratora.
    - Parametry: int id
    - Zwraca: Przekierowanie do listy gatunków.
- ListBooksByGenre (GET):
    - Wyświetla listę 10 książek w danym gatunku posortowaną według ocen malejąco. Dostępne dla administratora.
    - Parametry: int genreId, int page = 1, int pageSize = 10
    - Zwraca: Widok z listą książek.

4. HomeController

Kontroler zarządzający wyszukiwaniem oraz stroną błędu.

- Search (GET):
    - Wyszukuje książki lub autorów, w zależności od wybranej funkcjonalności przekazanej parametrem searchType. Tworzy ViewModel AuthorBookSearchResultsView zawierający listy AuthorViewModel oraz BookViewModel. Zawiera paginację. Zapisuje searchQuery i searchType w celu odczytu na następnych stronach.
    - Parametry: string searchQuery, string searchType, int page = 1, int pageSize = 10
    - Zwraca: Widok SearchResults.cshtml wraz z obiektem AuthorBookSearchResultsView.

- Error (GET):
    - Wyświetla stronę błędu.
    - Zwraca: Widok błędu.

5. UserLibraryController

Kontroler zarządzający biblioteką użytkownika.

- AddToLibrary (POST):
    - Dodaje książkę do biblioteki użytkownika. Jeśli książka już istnieje w bibliotece użytkownika, aktualizuje jej status. Dostępne dla zalogowanych użytkowników.
    - Parametry: int bookId, Status status
    - Zwraca: Przekierowanie do poprzedniej strony.
- DeleteFromLibrary (POST):
    - Usuwa książkę z biblioteki użytkownika. Dostępne dla zalogowanych użytkowników.
    - Parametry: int bookId
    - Zwraca: Przekierowanie do poprzedniej strony.
- Index (GET):
    - Wyświetla bibliotekę użytkownika. Dostępne dla zalogowanych użytkowników.
    - Parametry: int page = 1, int pageSize = 10
    - Zwraca: Widok z biblioteką użytkownika.

6. PaginationController

Kontroler bazowy zarządzający paginacją.

- SetPaginationData:
    - Ustawia dane paginacji i przekazuje do ViewData, które jest odczytywane przez widok częściowy Pagination.cshtml.
    - Parametry: int currentPage, int totalPages

Opis systemu użytkowników

W systemie funkcjonują dwie role użytkowników:

- User: Standardowy użytkownik aplikacji.
- Admin: Administrator z dodatkowymi uprawnieniami.

Każde nowe konto ma przypisaną rolę „User”. Zalogowani użytkownicy mogą zarządzać książkami w swojej bibliotece, śledzić ulubionych autorów, oraz dodawać recenzje i oceny książek.

Konto administratora ma wszystkie powyższe uprawnienia, oraz możliwość tworzenia, modyfikowania oraz usuwania książek, autorów i gatunków. Konto administratora jest tworzone automatycznie przy uruchomieniu aplikacji:

- Login: admin@admin.com
- Hasło: 123123

Krótka charakterystyka najciekawszych funkcjonalności

1. Wczytywanie datasetu

   Klasa DatasetLoader automatycznie ładuje dane początkowe z pliku CSV do bazy danych podczas pierwszego uruchomienia, co ułatwia rozwój aplikacji.

2. Recenzje książek

   Użytkownicy mogą dodawać recenzje do książek oraz przeglądać recenzje innych użytkowników. Recenzje zawierają tekst oraz ocenę. Ocena zostaje dodana do ogólnej średniej ocen danej książki. Każdy użytkownik może dodać tylko jedną recenzję do konkretnej książki. Funkcjonalność zaimplementowana w kontrolerze BooksController za pomocą metody AddReview().

3. Śledzenie autorów

   Użytkownicy mogą śledzić swoich ulubionych autorów. Metoda Index() w kontrolerze AuthorsController wyświetla widok skupiający wszystkich śledzonych autorów wraz z ich najnowszą książką.

4. Widoki częściowe

   Projekt wykorzystuje widoki częściowe do renderowania powtarzalnych elementów interfejsu takich jak lista książek _BooksList.cshtml i Book.cshtml, co zapewnia jednolity wygląd niezależnie od wyświetlonego widoku.

5. Paginacja

   Projekt obsługuje paginację książek oraz autorów, co pozwala na przyśpieszenie ładowania wyników.